



**UNIVERSIDADE DO MINHO
DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR**

CURSO DE LICENCIATURA EM INFORMÁTICA DE GESTÃO

RELATÓRIO DE PROJECTO LICENCIATURA

**SISTEMA MARCAÇÃO DE CONSULTA E ANÁLISE ONLINE
ANO LETIVO 2017-2018 – 4º ANO**

Autora: Joana Baptista Pires Neves, N.º 2847

Orientador: Mestre Paulo Santos Silva

Mindelo, 2018

JOANA BAPTISTA PIRES NEVES

**DESENVOLVIMENTO DE UM
SISTEMA DE MARCAÇÃO DE
CONSULTA E ANÁLISE
ONLINE**

Universidade do Mindelo

Mindelo, 2018

Joana Baptista Pires Neves

Mindelo, 2018

UNIVERSIDADE DO MINDELO

DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR

**CURSO DE LICENCIATURA
EM INFORMÁTICA DE GESTÃO**

RELATÓRIO DE PROJETO DE LICENCIATURA

Ano letivo 2017-2018 – 4º Ano

Autora: Joana Baptista Pires Neves, N.º 2847

Mindeló, 2018

Joana Baptista Pires Neves

SISTEMA MARCAÇÃO DE CONSULTA E ANÁLISE ONLINE

Trabalho de Conclusão de Curso
apresentado à **Universidade do Mindelo**
como parte dos requisitos para obtenção do
grau de licenciatura em **Informática de
Gestão**.

Orientador: Mestre Paulo Santos Silva

DEDICATÓRIA

Dedico este trabalho a Deus por me dar força e ter me ajudado a não desistir diante das dificuldades apresentadas.

Aos meus pais, Alcides Santos Neves e Virgínia Francisca Pires, pela educação, compreensão, carinho e paciência nos momentos mais difíceis.

Amo-vos de todo o meu coração.

AGRADECIMENTOS

A realização deste trabalho só se tornou possível dada a colaboração e contributo de várias pessoas a quem não posso deixar de agradecer.

Primeiramente agradeço a Deus, por me ter acompanhado e fazer-me acreditar que nada é impossível quando mantemos a fé.

Aos meus pais Alcides Neves e Virgínia Pires, aos meus irmãos que eu amo muito, pela partilha, compreensão, carinho, força suporte emocional, que ao longo de todo o percurso me proporcionaram.

Deixo um especial agradecimento ao meu namorado, Jorge Medina, por contribuir com o seu amor, carinho, compreensão e conselhos para que o foco fosse mantido e o objetivo alcançado.

A todos os professores pelos conhecimentos partilhados e em especial ao meu orientador da Monografia, Mestre Paulo Santos Silva, pela disponibilidade que sempre manifestou e pelas sugestões teóricas e metodológicas, pelo carinho, atenção, paciência e em que muito contribuíram para a realização do trabalho.

Aos colegas do curso, com os quais compartilhei momentos inesquecíveis, em especial a Carlina Pinto e Délcly Coronel que foram amigas verdadeiras.

Á todos os familiares e amigos que me deram força nos momentos mais difíceis principalmente durante a realização do trabalho.

Por fim quero fazer um agradecimento especial ao meu amigo Armindo Ferreira, Eurides Costa, Emerson Santos e ao FPE (Fundo Perpetuo de Educação) sem dúvida foram muito importantes para o término do curso, um muito obrigado do fundo do meu coração.

E a todos que contribuíram direta ou indiretamente para a realização desse trabalho.

“Suba o primeiro degrau com fê.
Você não tem que ver toda a escada.
Você só precisa dar o primeiro passo.”
(Martin Luther King)

RESUMO

Este trabalho tem como objetivo apresentar um projeto para o desenvolvimento de um sistema de marcação de consulta e análises online. Perante as dificuldades sociais, em relação a locomoção e tempo despendido na fila são fatores que podem ser minimizados através de um sistema que gere todo o processo de marcações dos serviços (Consulta, Analise e Tratamentos).

A realização dos respetivos serviços visa tornar virtual cada solicitação através do uso de plataformas utilizando as tecnologias apropriadas para tal.

O tempo despendido na fila para a marcação de consultas nos hospitais públicos ou nos centros de saúde são dificuldades inerentes a superlotação das urgências desses serviços. Entretanto o sistema SIMARC (Sistema de Marcação de Consultas e Análises online), veio na ótica de colmatar ou minimizar essas situações de desconforto de estar durante várias horas numa fila e não só como também descongestionar bastante o serviço de atendimento ao público.

Este projeto começa-se por apresentar uma breve introdução onde se inclui os objetivos gerais e específicos, a justificativa, a motivação pela escolha do tema, e a metodologia utilizada para a sua realização. Em seguida, passa-se para a fundamentação teórica relativamente ao tema do projeto em estudo e depois procede-se à modelação do sistema onde é apresentado os diagramas de case use, diagrama de classe, diagramas de sequências e dicionários de dados. Por fim o desenvolvimento do projeto, suas ferramentas e tecnologias e também as interfaces do sistema desenvolvido.

O sistema foi desenvolvido na linguagem PHP, com o auxílio de Yii2 framework juntamente com o adminLTE e a base de dados criada em MySQL, utilizando o sistema de gestão de base de dados (SGBD), *Phpmyadmin*.

Palavras-Chave: Sistema de Marcação de Consulta e Analise Online (SIMARC), PHP, Base de Dados, Yii2 framework.

ABSTRACT

This work aims to present a project for the development of an online appointment marking system and medical exams. Standing before the social difficulties, in relation to locomotion and time spent in the queue are factors that can be minimized through a system that manages the entire service process (Consults, medical exams and Treatment).

The accomplishment of the respective services aims to make every virtual request through the use of platforms using the proper technologies for such.

The time spent in line for making appointments in public hospitals or health centres are difficulties inherent in overcrowding the urgencies of these services. However, the system SIMARC (System appointment marking and Online Analysis), came in the perspective of filling or minimize these situations of discomfort of being for several hours in a queue and not only as well as also decongest the service of attendance to the public.

This project starts by presenting a brief introduction that includes the general and specific objectives, the justification, the motivation for choosing the theme, and the methodology used for its accomplishment. Then, we move on to the theoretical basis for the subject of the project under study and then proceed to the design of the system where it presents the case use diagrams, class diagram, sequence diagrams and data dictionaries.

In conclusion, the development of the project, its tools and technologies and also the interfaces of the developed system.

The system was developed in PHP language, with the support of Yii2 framework together with the adminLTE and the database created in MySQL, using the database management system (DBMS), Phpmyadmin.

Key words: *online appointment marking and medical exams (SIMARC), PHP, Database, Yii2 framework.*

Índice

CAPITULO I.....	1
1. Introdução	1
1.1. Contextualização	1
1.2. Motivação.....	2
1.3. Objetivos	2
1.3.1. Geral.....	2
1.3.2. Específicos	2
1.4. Justificativa	3
1.5. Metodologia	4
1.6. Estrutura do trabalho.....	5
CAPITULO II.....	6
2. Fundamentos Teórico	6
2.1. Sistemas de Informação	6
2.1.1. Componentes de um sistema de informação	8
2.1.2. Sistemas de Informação e as Organizações.....	8
2.1.3. Objetivo de um Sistema de Informação.....	9
2.1.4. Vantagens e Desvantagens de Sistemas de Informação.....	9
2.2. Sistemas de informação em saúde.....	10
2.2.1. Papel de um sistema de informação em saúde.....	11
2.3. Tecnologia WEB	11
2.3.1. Historico da internet e da web	12
2.3.2. Uma plataforma de acesso a sistemas de informação.....	13
2.3.3. Sistemas e Aplicações Web	14
2.3.4. Sistemas Web, vantagens e desvantagens.....	15
2.4. Ferramentas para Implementação do Protótipo.....	16
2.4.1. Yii Framework.....	17
2.4.1.1. Segurança do Yii	18
2.4.1.2. Arquitectura do Yii.....	18
2.4.1.3. Camada de modelo (Model)	18
2.4.1.4. Camada de controlo (Controller).....	19
2.4.1.5. Camada de visualização (View).....	20
2.4.1.6. Widgets	20
2.4.1.7. Assets.....	21

2.4.1.8.	Forms	22
2.4.1.9.	MCV no PHP	22
2.4.1.10.	AdminLTE.....	23
2.4.2.	PHP.....	24
2.4.3.	HTML.....	26
2.4.4.	JQuery.....	27
2.4.5.	CSS.....	27
2.4.6.	Base de Dados MySQL.....	28
2.4.7.	MySQL Workbench	28
2.4.8.	UML	29
2.4.9.	Apache.....	29
2.4.10.	Bootstrap	30
2.4.11.	JavaScript	30
CAPITULO III.....		32
3.	Modelação do Sistema	32
3.1.	Análises de Requisitos do Sistema	32
3.1.1.	Requisitos funcionais	32
3.1.2.	Requisitos não funcionais	33
3.2.	Desenho do Sistema.....	33
3.2.1.	Diagramas de caso de uso	34
3.2.2.	Diagrama de classes	36
3.2.3.	Diagrama de sequencia	37
3.2.4.	Dicionário de dados	38
CAPITULO IV		52
4.	Implementação do Protótipo	52
4.1.	Ferramentas e tecnologias	52
4.2.	Descrição das funcionalidades do sistema.....	55
4.2.1.	Front-End do sistema.....	55
4.2.2.	Back-end do sistema.....	58
CAPITULO V.....		67
5.	Conclusão.....	67
5.1.	Dificuldades encontradas	67
5.2.	Trabalhos Futuros	68
Referências Bibliográficas		69

LISTA DE FIGURAS

Figura 1: Diagrama dos Componentes do Sistema	8
Figura 2: Utilização da tecnologia web como plataforma de acesso a outros SI	14
Figura 3: Diagrama da Estrutura Estática de App no Yii	23
Figura 4: Tela do AdminLTE	24
Figura 5: Diagrama caso uso Administrador	34
Figura 6: Diagrama caso uso do Paciente	35
Figura 7: Diagrama caso uso do Medico	35
Figura 8: Diagrama de Classe do sistema.....	36
Figura 9: Diagrama sequencia Fazer Login	37
Figura 10: Diagrama sequencia Registrar Paciente	37
Figura 11: Exemplo de um Código PHP	52
Figura 12: Exemplo de um Código HTML	53
Figura 13: Base de Dados do Sistema	54
Figura 14: Exemplo de um Código CSS.....	55
Figura 15: Tela Página Inicial.....	56
Figura 16: Tela Página Simarc	57
Figura 17: Tela Página Consulta.....	57
Figura 18: Tela Página Contato	58
Figura 19: Tela login do sistema	59
Figura 20: Tela Principal do sistema	59
Figura 21: Tela Menus do sistema	60
Figura 22: Tela Cadastrar Funcionário do sistema	60
Figura 23: Tela Agendar Consulta no sistema.....	61
Figura 24: Tela Marcar consulta	61
Figura 25: Tela Paciente em espera no sistema	62
Figura 26: Tela Consulta no sistema	62
Figura 27: Tela Lista Pacientes do sistema.....	63
Figura 28: Tela Cadastrar Pacientes no sistema	63
Figura 29: Tela Lista Domínios do sistema	64
Figura 30: Tela Adicionar Domínios do sistema	64
Figura 31: Tela Lista de Role do sistema	65
Figura 32: Tela Criar Role do sistema	65
Figura 33: Tela Criar Routes do sistema	66

LISTA DE TABELAS

Tabela 1: Estrutura do Relatório	5
Tabela 2: Dicionário de dado da tabela Analise	38
Tabela 3: Dicionário de dado da tabela Consulta	39
Tabela 4: Dicionário de dado da tabela Bolitim_Clínico	39
Tabela 5: Dicionário de dado da tabela Paciente	40
Tabela 6: Dicionário de dado da tabela Receita	40
Tabela 7: Dicionário de dado da tabela Calendário	40
Tabela 8: Dicionário de dado da tabela Complemento	41
Tabela 9: Dicionário de dado da tabela Contato	41
Tabela 10: Dicionário de dado da tabela Domínio	42
Tabela 11: Dicionário de dado da tabela Empresa	42
Tabela 12: Dicionário de dado da tabela Especialidade	42
Tabela 13: Dicionário de dado da tabela Geografia	43
Tabela 14: Dicionário de dado da tabela Ilhas	43
Tabela 15: Dicionário de dado da tabela Login	44
Tabela 16: Dicionário de dado da tabela LoginValidation	44
Tabela 17: Dicionário de dado da tabela Medicamento	45
Tabela 18: Dicionário de dado da tabela Medicamento_Receita	45
Tabela 19: Dicionário de dado da tabela Medico_Especialidade	45
Tabela 20: Dicionário de dado da tabela Menu	46
Tabela 21: Dicionário de dado da tabela MenuPerfil	46
Tabela 22: Dicionário de dado da tabela Pais	46
Tabela 23: Dicionário de dado da tabela Perfil	47
Tabela 24: Dicionário de dado da tabela Resultado_Analise	47
Tabela 25: Dicionário de dado da tabela Especialidades	48
Tabela 26: Dicionário de dado da tabela Tipo_Analise	48
Tabela 27: Dicionário de dado da tabela Horario_Consulta	48
Tabela 28: Dicionário de dado da tabela Tratamento	49
Tabela 29: Dicionário de dado da tabela Documentos_Pacientes	49
Tabela 30: Dicionário de dado da tabela Dados_Clínicos	50
Tabela 31: Dicionário de dado da tabela Migration	50
Tabela 32: Dicionário de dado da tabela Auth_Assignment	50
Tabela 33: Dicionário de dado da tabela Auth_Item	51
Tabela 34: Dicionário de dado da tabela Auth_Item_Child	51
Tabela 35: Dicionário de dado da tabela Auth_Rule	51

LISTA DE ACRÓNIMOS E SIGLAS

BD	Base Dados
CRUD	Create – Read – Update – Delete
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DD	Dicionário de Dados
HTML	Hyper Text Mark Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
MVC	Model – View – Controller
PHP	Hypertext Preprocessor
POO	Programação Orientada a Objetos
SGBD	Sistema Gestão Base Dados
SI	Sistema de Informação
SIS	Sistema de Informação em Saúde
TI	Tecnologia de Informação
UML	Unified Modeling Language
URL	Uniform Resource Locator
WWW	World Wide Web
XML	Extensible Markup Language

CAPITULO I

1. Introdução

O trabalho que por hora se apresenta, tem como objetivo fundamental a criação e o desenvolvimento de um sistema de marcação de consultas e análises online. Como sabemos nos dias de hoje torna-se imprescindível o uso de novas tecnologias de comunicação e informação em quase tudo ou se não em tudo o que seja o nosso dia-a-dia, isto é, nas escolas, nas organizações empresariais, na ciência, na governação e principalmente no sector de saúde.

1.1. Contextualização

Com o passar dos tempos, devido ao desenvolvimento tecnológico neste sector, viu-se a esperança média de vida das pessoas do mundo inteiro a aumentar consideravelmente, pois houve desenvolvimento exponencial na fabricação de muitos medicamentos, mais eficazes no tratamento de várias doenças, melhorias nos procedimentos hospitalares e excelentes aparelhos tanto nos diagnósticos como também nos respetivos tratamentos. Dito isto, também convém realçar que em todo mundo têm-se desenvolvido vários mecanismos com o intuito de reduzir o tempo de espera e a necessidade de acabar com a obrigatoriedade da presença dos utentes do sistema nacional de saúde.

Perante tais necessidades abraçou-se o desafio da criação e desenvolvimento do sistema SIMARC. Sistema esse que vem incorporando diretamente a valência de marcações online de consultas e análises, pois, demonstra ser de capital importância para os utentes que já a partir de um computador, tablet ou smartfone podem fazer a sua marcação sem sair da comodidade da sua casa e sem que esteja nas longas filas de espera para fazer o mesmo, perdendo imenso tempo que poderia ser empregue noutros afazeres. Também para os hospitais essa plataforma pode-se apelidá-lo de mais uma ferramenta de gestão eficaz, simples e rápido no desempenho das funções a que foi criado. Poupa-se algum dinheiro na contratação de pessoas que trabalham nestes sítios. Como uma importante ferramenta de gestão de consultas e análises pode-se afirmar que inovar nesta vertente atualmente é imperativo para os hospitais, centros de saúde, clínicas públicas e privadas em que com certeza será um investimento inicial que dará os seus frutos. O uso das

aplicações Web veio revolucionar e de que maneira, melhorar os suportes de gestão em diversas áreas inclusive na saúde.

1.2. Motivação

O tema escolhido para realização deste projeto, surgiu devido a necessidade de fomentar e criar um sistema de marcação de consultas online que facilita uma melhor gestão dos hospitais, centros de saúde e dos utentes utilizando as tecnologias para tais serviços. Além disso, o tema surgiu perante a necessidade de concretizar os objetivos propostos ao longo do curso e releva uma perspetiva de realização pessoal e académica, uma vez que, almejou-se obter conhecimentos mais aprofundados a respeito do tema.

A pressão que a sociedade atual sofre devido a evolução tecnológica, acredita-se que a implementação desse sistema seja de grande relevância e que venha contribuir para a compreensão e reconhecimentos dos seus benefícios, facilitando a vida dos utentes, pois é rápido, prático e de fácil acesso, basta estar conectado a internet.

1.3. Objetivos

O objetivo principal deste trabalho é implementar um sistema de marcação de consulta e análise online. Pretende-se com este projeto, consolidar e aumentar os conhecimentos em gestão de informação, programação em diferentes linguagens.

1.3.1. Geral

Projetar e construir um protótipo que permita fazer uma marcação de consulta e análise online.

1.3.2. Específicos

Os objetivos específicos do trabalho são:

- a) Análise do sistema a desenvolver e melhorias na área de gestão de TI;
- b) Agilizar o processo de marcação de consultas e análises de forma automatizada;

- c) Conceção do sistema recorrendo ao software aplicação do sistema de marcação de consultas e análises;
- d) Realização de levantamento de requisitos necessários para desenvolvimento;
- e) Realização de testes e implementação do sistema.

1.4. **Justificativa**

O atual processo de agendamento de uma simples consulta, sendo feito somente com a presença física do utente, conduz a fatores que dificultam e causam transtornos. A sociedade têm-se tornado cada vez mais competitiva e todos os recursos disponíveis têm de ser devidamente controlados. Com o avanço das tecnologias vem requerendo cada vez mais sistemas sofisticados, modernos e que vão de encontro às expectativas dos gestores dos recursos humanos.

É neste contexto que se propõe o projeto, construção e implementação de um sistema de marcação de consulta e análise online, como projeto final de licenciatura em Informática de Gestão.

Sendo assim, viu-se na necessidade de desenvolver um sistema de marcação de consultas e análises de forma a facilitar os utentes de determinados constrangimentos, tais como, o desconforto de estar várias horas numa fila a espera para ser atendido, o constrangimento de se levantar muito cedo em que muitas vezes o utente nem se quer terá garantias que será atendido devido á imensa afluência de pessoas aos serviços. Este sistema deve permitir que os utentes possam marcar as suas consultas ou análises, permitir que o administrador do sistema possa gerenciá-lo, bem como os médicos e também permitir validar as marcações relativamente as consultas e análises.

O projeto de sistema de marcação de consulta e análise será elaborado no geral utilizando a linguagem programação PHP, juntamente com HTML e com o sistema de gestão de base de dados MYSQL utilizando a ferramenta de desenvolvimento yii2 framework, terá

como principais funcionalidades fazer a marcação das consultas e análises dos utentes. Para além disso, o sistema deverá ser capaz de registar novos utentes, gerar relatórios, permitir fazer pesquisas de consultas e análises. Este terá uma interface de gestão e consulta *web-base* através de yii2 framework.

1.5. Metodologia

Nesta fase, descrevem-se os procedimentos a seguir e as técnicas a serem utilizadas nas diferentes etapas da investigação.

Para desenvolver-lo foi preciso várias pesquisas sobre aplicações web, engenharia de *softwares*, com o intuito de entender a especificação, desenvolvimento, manutenção e criação de sistemas de *software*, com aplicação de tecnologias e práticas de gestão de projetos e outras disciplinas, visando organização, produtividade e qualidade. Estas tecnologias e práticas englobam linguagens de programação, base de dados, ferramentas, plataformas, bibliotecas, padrões, processos e a questão da qualidade de *software*.

Na elaboração de uma pesquisa, propõem-se a escolha de um tema e se defina um problema para ser investigado, elabore um plano de trabalho e após a execução do plano, escreve-se um relatório final e que esteja apresentado de forma planificada, ordenada, lógica e conclusiva.

Na elaboração desta monografia académica, a metodologia foi dividida em cinco fases desde a pesquisa bibliográfica documental, em sítios web especializados no assunto apresentado e vários outros meios de informação que ajudaram a compreender melhor o assunto e definir o projeto.

As fases de concessão do projeto:

- Definição dos objetivos, bem como o planeamento das fases do projeto e suas atividades;
- Estudo sobre sistemas baseados na tecnologia e as ferramentas a serem utilizadas ao longo da elaboração do trabalho;
- Análise e modelação dos dados para assegurar a completa e correta definição das funcionalidades do sistema e a satisfação das necessidades dos utilizadores;

- Desenvolvimento do protótipo através das linguagens PHP, HTML, CSS juntamente com a elaboração da base de dados MySQL, utilizando o JavaScript para tornar o site dinâmico e a ferramenta Bootstrap para tornar a interface gráfica mais atrativa;
- Por fim, na última fase, foram tiradas as conclusões, descritos os resultados e feitas sugestões para desenvolvimentos futuros.

1.6. Estrutura do trabalho

O presente trabalho encontra-se organizado em cinco capítulos, em que no primeiro fez-se uma breve introdução, a contextualização, a motivação, a justificativa, a metodologia utilizada e a estrutura do mesmo. No segundo e terceiro capítulos é apresentado todo o enquadramento teórico que sustenta a parte prática. No quarto capítulo estão esplanados os aspetos técnicos referentes ao seu desenvolvimento, descrevendo as suas especificidades da sua implementação, entre as quais o planeamento, as tecnologias e ferramentas utilizadas. No quinto capítulo encontra-se a conclusão e as considerações finais.

CAPITULO I	INTRODUÇÃO
CAPITULO II	FUNDAMENTOS TEÓRICOS
CAPITULO III	MODELAÇÃO DO SISTEMA
CAPITULO IV	DESENVOLVIMENTO DO SISTEMA
CAPITULO V	CONCLUSÃO E CONSIDERAÇÕES FINAIS

Tabela 1: Estrutura do Relatório

Fonte: Elaborado pela autora desse trabalho

CAPITULO II

2. Fundamentos Teóricos

Este capítulo, tem por objetivo a apresentação dos principais conceitos teóricos necessários ao desenvolvimento deste trabalho. Serão apresentados alguns conceitos básicos sobre sistemas de informação, bem como as ferramentas utilizadas ao longo do desenvolvimento desse projeto.

2.1. Sistemas de Informação

Segundo Xexéo (2006), Sistemas de Informação são utilizados em organizações para planejamento, monitorização, comunicação e controle das suas atividades, por meio da manipulação e guarda de informações.

Antes de desenvolver qualquer conteúdo é de extrema importância conhecer o seu conceito para poder entender melhor o que venha ser feito. Tendo em conta tal, segue-se alguns conceitos mais importantes que nos ajudam no melhor entendimento do tema em estudo.

Para Ferreira (2008) **Sistema de Informação** é um tipo especializado de sistemas que pode ser definido como sendo um conjunto de elementos ou componentes inter-relacionados que coletam (entrada), manipulam e armazenam (processo), disseminam (saída) os dados e informação e fornecem um mecanismo de feedback.

Para Xexéo (2006) **Sistema de Informação** é um conjunto de elementos inter-relacionados que coleta dados no ambiente em que opera, analisa esses dados e apresenta o produto final trabalhada de forma a atender as necessidades de uma organização, de seus clientes internos e externos e de todos aqueles que atingidos direta ou indiretamente pelo novo produto final.

Segundo Rascão (2004, p. 26) “**Sistemas de Informação** é um conjunto organizado de procedimentos, que, quando executados, produzem informação para o apoio a tomada de decisão e ao controlo das organizações”. Partilhando a mesma ideia sobre sistemas de

informação, pode-se dizer que Sistemas de Informação é um conjunto de independentes (subsistemas), logicamente associados, para que de sua interação sejam geradas informações à tomada de decisão, e que sistema pode ou não envolver tecnologias de informação.

Os sistemas de informações podem ser classificados a princípio como formais ou informais. Os sistemas de informação formais incluem processo pré-definidos, entrada e saídas padronizadas e definições fixas. Quanto as informais, estes assumem diversas formas, que vão desde uma rede de comunicação informal em uma empresa, até um grupo de amigos que troca correspondência eletronicamente.

Um Sistema de Informação contém informação sobre uma organização e seu ambiente, além do suporte à tomada de decisão, coordenação, controle, os sistemas de informação auxiliam gerentes e funcionários a analisar problemas, visualizar soluções e também criar novos produtos.

Atividades básicas que produzem as necessidades de informação da organização:

- Entrada;
- Processamento;
- Saída;

2.1.1. Componentes de um sistema de informação

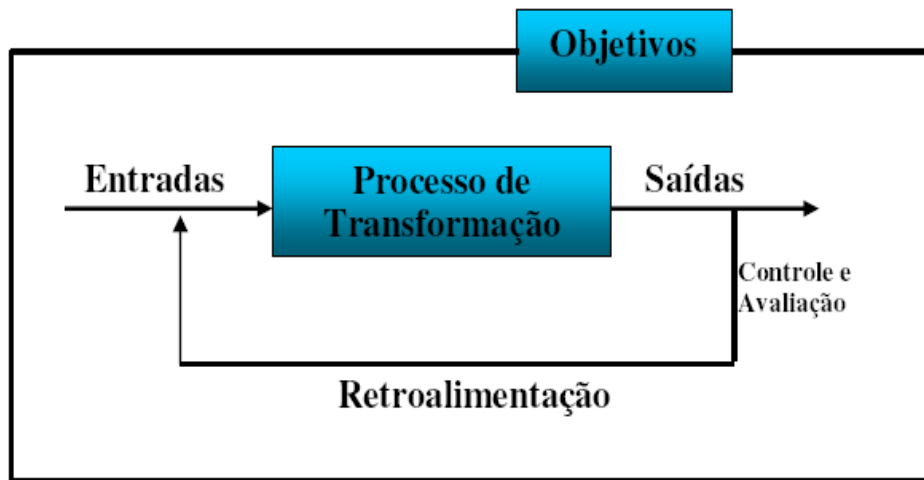


Figura 1: Diagrama dos Componentes do Sistema

Fonte:

<https://www.google.com/search?q=Componentes+de+um+sistema+de+informa%C3%A7%C3%A3o&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwiF5obB5vPbAhUEZ8AKHYXFA6IQsAQIWw#imgsrc=Ncy2dlhBnzRFzM>

2.1.2. Sistemas de Informação e as Organizações

Atualmente as empresas precisam estar centradas em suas informações, exigindo de seus gestores estratégias eficientes, que podem ser facilitadas utilizando-se de recursos inteligentes oferecidos pela tecnologia de informação e sistemas de informação gerencial. A necessidade de que as organizações sejam dinâmicas, diante das mudanças constantes da sociedade de informação, faz com que elas também se modifiquem e requeiram planejamento de suas informações auxiliadas pelos recursos das Tecnologias.

Os avanços tecnológicos estão a obrigar as organizações a mudanças de paradigma e adaptações, e com isso traz benefícios tais como:

- Redução ou substituição de mão-de-obra;
- Aumento da capacidade dos colaboradores;
- Apoio na tomada de decisão;
- Substituição de papéis;
- Coordenação de processo à distância;
- Redução dos custos;

- Satisfação dos clientes.

2.1.3. Objetivo de um Sistema de Informação

O aumento crescente de volume de dados é um reflexo direto nos sistemas de informação e da sociedade em modo geral. Com isso nasce a necessidade de criar um sistema de informação, capaz de recolher, armazenar e processar dados para atingir os objetivos.

Segundo RASCÃO (2004, p. 27) um Sistema de Informação é constituído por vários subsistemas que têm diferentes funções, tentando alcançar os seguintes objetivos:

- Recolher, selecionar, tratar e analisar os dados capazes de ser transformado em informação;
- Proporcionar, regularmente a informação operacional requerida pelos gestores operacionais de forma a assegurar o funcionamento do dia-a-dia da organização;
- Proporcionar de forma regular os gestores de nível intermédio, de modo a coordenar as atividades da sua área de responsabilidade, para atingir os objetivos e tomarem as decisões de corrigir os eventuais desvios;
- Proporcionar, de forma regular ou irregular a informação aos gestores de nível intermédio e de topo, de modo a permitir-lhes tomar as melhores decisões a cerca do futuro da organização;
- Acrescentar valor à organização. Isto significa que o sistema de informação da organização se relacionará com sistemas de informação externos.

2.1.4. Vantagens e Desvantagens de Sistemas de Informação

Os sistemas de informação são muito importantes para uma melhor circulação da informação nas organizações e desta forma uma importantíssima ferramenta para o crescimento das mesmas.

Organizações bem-sucedidas, atualmente, apresentam um sistema de informação sofisticado que é na verdade a principal responsável para a obtenção de tais êxitos nas suas atividades que desenvolvem. Toda organização necessita de informação que seja benéfica, útil e de baixo custo, acessível, organizado e preciso. Portanto, sistemas de informação deve ser constantemente mantido e atualizado para atender as expectativas da empresa e as necessidades dos clientes.

As suas principais vantagens de um SI bem construído:

- Optimizar o fluxo da informação e a qualidade da mesma dentro da organização (eficiência);
- Optimizar o processo de tomada de decisão;
- Eliminar a redundância de atividades;
- Reduzir as incertezas do lead-time (tempo de aprovisionamento);
- Acessar, compreender e responder às informações de forma rápida e eficaz;
- Redução de custos operacionais e administrativos e ganho de produtividade;
- Maior integridade e veracidade da informação;
- Maior segurança de acesso à informação;

Desvantagens

- A utilização do ERP (Enterprise Resource Planning);
- Excesso de controlo sobre as pessoas, o que aumenta a resistência à mudança e pode-se gerar desmotivação por parte dos funcionários;
- Nem sempre o sistemas funcione corretamente, devido a diversos fatores;
- Quando ocorre defeitos no sistema, as informações que advem deste, de maneira alguma serão confiáveis para o seu uso ou ponderamento para tomadas de decisões;
- Os sistemas são vulneráveis a hackers e fraudes.

2.2. Sistemas de informação em saúde

Sistema de Informação em Saúde -SIS como um mecanismo de coleta, processamento, análise e transmissão da informação necessária para se planejar, organizar, operar e avaliar os serviços de saúde. Considera-se que a transformação de um dado em informação exige, além da análise, a divulgação, e inclusive recomendações para a ação.

De acordo com ARROYO et OLIVEIRA (2004) referido por Flávia Fadel (2007), O avanço da medicina pode ser enriquecido com a introdução dos dados do paciente em meio eletrônico, com a inclusão de sistema de apoio à decisão para diagnóstico e prescrição de medicamentos e com a utilização de base de dados sobre casos relacionados,

dentre outros, tudo isto consequência da introdução de Sistema de Informação (SI) e Tecnologias de Informação (TI).

2.2.1. Papel de um sistema de informação em saúde

- Organizar a produção de informações compatíveis com as necessidades dos diferentes níveis, garantindo uma avaliação permanente das ações executadas e do impacto destas sobre a situação de saúde;
- Assegurar o desenvolvimento de sistemas voltados para as especificidades das diferentes unidades operacionais do sistema de saúde;
- Contribuir para o desenvolvimento dos profissionais de saúde, para a construção de uma consciência sanitária coletiva, como base para ampliar o exercício do controle social e da cidadania.

2.3. Tecnologia WEB

A *World Wide Web*, *WWW*, ou simplesmente *Web*, foi desenvolvida para ser “um pool” de conhecimento humano, que permitisse que as pessoas estando em locais distantes umas das outras, mesmo assim pudessem partilhar ideias e documentos.

A tecnologia *Web* pode ser definida como um sistema de padrões que inclui:

- Padrão de endereçamento: todos os recursos da *Web* têm um endereço único e podem ser localizados de qualquer parte, independentemente da plataforma onde o recurso resida. Cada endereço é chamado de URL (*Uniform Resource Locator*);
- Padrão de comunicação: a tecnologia web utiliza um protocolo de comunicação, ou seja, uma linguagem que permite a solicitação e obtenção de recursos da *Web*. Este protocolo, chamado HTTP (*HyperText Transfer Protocol*), permite a busca de recursos em diversos formatos e não somente de hipertexto como o nome sugere;
- Padrão de estruturação das informações: o padrão inicial da tecnologia *Web* para apresentação das informações estava baseado em uma linguagem de marcação chamada HTML (*Hypertext Markup Language*). Esta linguagem define principalmente elementos para a visualização de informações. Entretanto, uma extensão da tecnologia web foi a definição da metalinguagem chamada XML

(*Extensible Markup Language*) a qual permite definir de forma extensível como uma informação pode ser estruturada;

- chamada HTML (*Hypertext Markup Language*): esta linguagem define principalmente elementos para a visualização de informações. Entretanto, uma extensão da tecnologia web foi a definição da metalinguagem chamada XML (*Extensible Markup Language*) a qual permite definir de forma extensível como uma informação pode ser estruturada.

A tecnologia *Web* funciona utilizando o paradigma cliente-servidor. Neste modelo de computação, o processamento é dividido, conforme o nome sugere, entre clientes e servidores. Os clientes solicitam serviços, os quais são executados por servidores.

Na web, os clientes são *softwares* genéricos, chamados de navegadores, que proporcionam a interface com o utilizador. Os navegadores entendem os padrões da tecnologia web e são responsáveis por transformar as solicitações dos utilizadores em pedidos aos serviços web. Estes últimos recuperam os recursos (páginas) solicitados e os retornam aos clientes, que os interpretam, formatam e disponibilizam aos utilizadores.

2.3.1. Historico da internet e da web

A Internet, surgiu em pleno período da guerra fria, isto é foi criada com objetivos militares, em que a sua principal função seria de manter as comunicações Norte-americanas, no caso de haver qualquer ataque inimigo.

Nas décadas de 1971 e 1981, a internet já era um importante meio de comunicação académica, pois os professores e alunos universitários já trocavam ideias e partilhavam informações nos Estados Unidos da América.

Entretanto apesar da sua criação, contar com um período de tempo bastante considerável, somente em 1991 que a internet começou a alcançar a população geral. Neste ano o engenheiro Inglês Tim Bernes-lee desenvolveu a world wide web (www), possibilitando a utilização de interface gráfica e criação de sites mais dinâmicos. A partir desse momento a internet ganhou uma enorme dinâmica de crescimento e expansão pelo mundo.

Desde então, esta tem sido cada vez mais importante para todos os cidadãos do mundo inteiro, visto que esta importante ferramenta tecnológica é a principal responsável pelo fenómeno da Globalização. Pode-se afirmar de que hoje é praticamente quase impossível viver sem a internet, porque o homem adaptou a satisfação das suas necessidades a soluções provenientes em grande parte desse meio.

2.3.2. Uma plataforma de acesso a sistemas de informação

Ao longo do tempo, novos recursos foram acrescentados à tecnologia web. Com eles tornou-se possível:

- Enviar, junto com uma solicitação, informações ao servidor;
- Guardar “estado” entre duas chamadas ao servidor;
- Realizar processamento simples no próprio navegador;
- Desviar uma solicitação para que possa ser processada em um aplicativo no servidor, possibilitando a montagem dinâmica de páginas web;
- Efetuar comunicações seguras entre clientes (navegadores) e os servidores.

Os novos recursos permitem que a tecnologia web seja utilizada como infraestrutura de acesso a sistemas de informação. Dessa forma, os utilizadores interagem com os sistemas através dos próprios navegadores web, fornecendo informações aos servidores, os quais processam e geram as respostas (páginas web) dinamicamente. Assim, a troca de informações entre utilizadores e web é bidirecional de forma similar ao que ocorre com os sistemas de informação baseados nas tecnologias tradicionais.

O Funcionamento da tecnologia web como plataforma para o acesso a sistemas de informação.

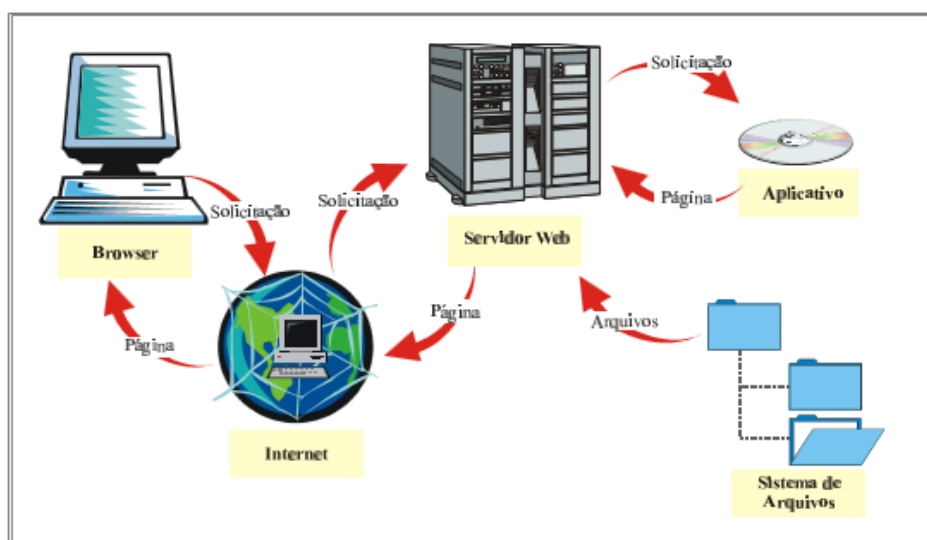


Figura 2: Utilização da tecnologia web como plataforma de acesso a outros SI

Fonte: Zeneti (2003)

Uma limitação para a utilização da tecnologia web, conforme mostrado acima, é que ela faz intermediação entre o navegador e o aplicativo, mas caso o aplicativo precise se comunicar com outro sistema ele deve utilizar uma tecnologia tradicional. Para contornar tal restrição. Novas extensões da tecnologia foram recentemente desenvolvidas, permitindo que seja usada também como infraestrutura de comunicação entre sistemas.

2.3.3. Sistemas e Aplicações Web

Segundo PRESSMAN (2002, p. 87) as seguintes características que guiam os sistemas baseados na Web são:

- **Imediatismo:** aplicações baseadas na Web têm um imediatismo que não é encontrado em nenhum outro tipo de *software*. Isto é, o prazo de colocação no mercado e disponibilização de novas informações de um Sítio Web pode ser uma questão de semanas. Os desenvolvedores precisam usar métodos para gerir todo o projeto;
- **Segurança:** como as aplicações Web estão disponíveis através de acesso à rede, é difícil limitar a população de utilizadores finais que podem ter acesso à aplicação. A fim de proteger o conteúdo reservado e fornecer modos seguros de transmissão de dados, boas medidas de segurança precisam ser implementadas na infra-estrutura da aplicação propriamente dita;

- **Estética:** uma inegável parte da atração de uma aplicação ou sistema Web é o seu aspeto. Quando uma aplicação é projetada para o mercado, para vender produtos e ideias, ou para que os utilizadores se sintam bem e fiquem à vontade no uso da aplicação a estética pode ter tanto a ver com o sucesso quanto o projeto técnico.

2.3.4. Sistemas Web, vantagens e desvantagens

Vantagens

A seguir estão citadas algumas vantagens da utilização dos sistemas que são baseados na plataforma Web (CONALLEN, 2003, p. 193):

- Sistemas na Web podem ser executados a partir de qualquer navegador da Internet;
- Sistemas baseados na Web podem ser acedidos de qualquer lugar do mundo, para isto basta apenas o utilizador possuir um computador com conexão à Internet e um navegador, pois o sistema Web fica acessível em qualquer computador seja na rede local ou na Web facilitando aos utilizadores o acesso ao aplicativo onde quer que estejam;
- Interface HTML reconhecida por uma grande gama de utilizadores já acostumados com o funcionamento dos navegadores;
- Desenvolvimento, manutenção e atualização centralizada da aplicação. Não é exigida muita memória e nem poderosos processadores para a execução do sistema nos terminais, pois o sistema é todo executado no servidor;
- Escalabilidade no processamento, se houver necessidade de aumentar o poder de processamento, basta fazer isto no servidor;
- Pode proporcionar potencialmente, uma melhor produção para a empresa, pois o funcionário pode trabalhar em qualquer local onde o acesso a Internet é disponibilizado.

Desvantagens

A seguir estão citadas algumas desvantagens da utilização dos sistemas que são baseados na plataforma Web (CONALLEN, 2003, p. 194):

- Não há uma padronização entre os diversos navegadores e o sistema poderia ser exibido de uma maneira diferente dependendo do navegador e da versão deste navegador também;
- A entrada de uma grande massa de dados é prejudicada na interface HTML, pois não existe uma maneira padrão de criar máscaras de entrada de dados;
- Tempo de processamento da execução das tarefas depende da velocidade da conexão, entre cliente e servidor;
- Os sistemas baseados na Web dependem dos recursos do navegador usado para visualizar a aplicação. Como eles possuem recursos diferentes, existem dificuldades para prever como a aplicação vai se comportar;
- Interface HTML não é rica em controlos gráficos e peca no quesito posicionamento;
- Dificil gestão do estado do cliente no servidor. A natureza sem conexão das comunicações do cliente e do servidor não proporciona um modo fácil do servidor controlar a solicitação de cada cliente e associá-la à solicitação anterior, visto que toda solicitação de página Web estabelecem e em seguida, interrompe um conjunto completamente novo de conexões.

2.4. Ferramentas para Implementação do Protótipo

Para desenvolvimento optou-se por usar o yii framework por ser uma das mais seguras na implementação de aplicações web. É também disponível gratuitamente tendo em conta que o custo das aplicações.

No Yii tudo é muito bem definido, as classes são de fácil extensão e compreensão. As funcionalidades que não estão no Yii são disponíveis em extensões. A grande vantagem é o uso do PHP que é uma linguagem de fácil acesso mesmo para iniciantes em programação, pois é livre de custos de manutenção e muito barato quanto a hospedagem.

O PHP está integrada na maioria das bases de dados da atualidade, é muito associado ao MYSQL mas pode rodar sem restrição no Oracle, MSSQL, IBM e outros. O yii é adaptado para desenvolvimento em PHP que é uma linguagem de fácil acesso, que se pode encontrar no desenvolvimento da maioria das aplicações web. Quanto a qualidade das aplicações web em PHP têm vindo a melhorar consideravelmente, principalmente

com a entrada no mercado do yii, facilitando e reduzindo o tempo de elaboração de qualquer sistema.

2.4.1. Yii Framework

Yii acrónimo para “Yes it is” é um framework Open Source de alta performance e escalabilidade em PHP, criado em 2008, que utiliza componentes para o desenvolvimento de grandes aplicações Web. Permite uma maior reutilização de códigos na programação Web e pode acelerar significativamente o processo de desenvolvimento. O nome Yii (pronunciado i) representa as palavras fácil (*easy*), eficiente (*efficient*) e extensível (*extensible*).

Para executar uma aplicação Web que utilize o Yii, precisamos de um servidor Web com suporte a PHP 5.1.1 ou superior. Para os desenvolvedores que desejam utilizar o Yii, é muito importante o conhecimento de programação orientada a objetos (POO), pois o Yii é um *framework* totalmente orientado a estes.

O Yii é um *framework* de programação Web genérico que pode ser usado para desenvolver praticamente todos os tipos de aplicações Web. Por ser um *framework* leve equipado com sofisticadas soluções em *caching*, é especialmente adequado para o desenvolvimento de aplicações com alto tráfego de dados, tais como portais, fóruns, sistemas de gestão de conteúdo (CMS), sistemas de *e-Commerce*, etc.

Como a maioria dos *frameworks* PHP, O Yii é um *framework* MVC. O Yii se sobressai dos outros *frameworks* PHP na medida em que é eficiente, rico em recursos e bem documentado. O Yii é cuidadosamente projetado para ajustar a sérias aplicações Web desde seu início.

Vantagem do Yii framewok

- É um excelente suporte a MVC, DAO / *Active Record*, I18N (*Internationalization*) / L11N (*Localization*);
- Autenticação e Controle de acesso (RBAC);
- Testes unitários, Gerador de códigos automáticos;
- *Skinning and Theming*, *Jquery* / solicitações Ajax integrado aos *widgets* do Yii;

- *Web Services* (WSDL) e muito mais;
- A sua arquitetura permite que o mesmo carregue somente o necessário para aplicação no presente momento e, em conjunto com o suporte a cache (APC), consegue um RPS (Requisição por Segundo).

2.4.1.1. Segurança do Yii

Por padrão faz o tratamento, validações de entrada e saída de dados, oferece suporte a ataques *SQL Injection*, *Cross-site scripting* (XSS), *CSRF* (*Cross-Site Request Forgery*), *Cookie Attack* e entre outros.

Aceita muito bem integrações com códigos de terceiros, como por exemplo, *PEAR*, *Codeigniter*, *Zend Framework*. Yii segue o padrão MVC em sua estrutura, garantindo a separação entre camadas lógicas e camadas de apresentação do projeto a ser desenvolvido. Auxilia o desenvolvedor a ter um código mais limpo e reutilizável (DRY – *Don't repeat yourself*) sem grandes esforços, supre a elaboração de sites simples bem como aplicações extremamente complexas, sendo necessário preocupar apenas com tarefas específicas do projeto.

2.4.1.2. Arquitetura do Yii

Para facilitar os desenvolvedores no tratamento de formulários, uma das principais tarefas no desenvolvimento, os criadores do Yii integraram a arquitetura MVC (*Model View Controller*).

A arquitetura MVC consiste na separação lógica das principais camadas da aplicação, desde a camada mais abstrata até a mais concreta. A proposta de separar as camadas parece muito fácil em sua teoria, entretanto, colocá-la em prática não é tão simples.

Sendo assim, os criadores do MVC padronizaram sua divisão basicamente em três camadas: modelo (*model*), visualização (*view*) e controlador (*controller*).

2.4.1.3. Camada de modelo (Model)

É a parte onde é feita a abstração dos dados. Neste é definido a forma e as restrições dos dados os quais estarão disponíveis na aplicação. O modelo interage diretamente com as

views e os controladores (camadas de visualização e controle, respetivamente). O modelo comunica a estes outros dois componentes quaisquer alterações feitas, com isso os controladores manipulam os dados e as visualizações renderizam.

O Yii 2.1 usa a `yii\base\Model` como modelo base, semelhante à `CModel` no 1.1. A classe `CFormModel` foi removida inteiramente. No Yii 2 estende-se a classe `yii\base\Model` para criar uma classe de modelo de formulário. O Yii 2.1 introduz um novo método chamado `scenarios()` para declarar os cenários suportados, e para indicar sob qual cenário um atributo precisa ser validado, pode ser considerado *safe* (seguro) ou não, etc. Por exemplo:

```
public function scenarios()
{
    return [
        'backend' => ['email', 'role'],
        'frontend' => ['email', '!name'],
    ];
}
```

No código acima, dois cenários são declarados: *backend* e *frontend*. Para o cenário *backend*, os atributos *email* e *role* são seguros (*safe*), e podem ser atribuídos em massa. Para o cenário *frontend*, *email* pode ser atribuído em massa enquanto *role* não. Tanto *email* quanto *role* devem ser validados utilizando-se *rules* (regras). O método `rules()` ainda é usado para declarar regras de validação. Perceba que devido à introdução do método `scenarios()`, não existe mais o validador *unsafe* (inseguro).

2.4.1.4. Camada de controlo (Controller)

A camada de controlo é responsável por manipular os dados vindos do modelo (*model*), além de enviá-los, possuindo alterações ou não de volta à camada de visualizações (*view*). Uma vez recebido o dado a ser manipulado, o controlador executa ações (*actions*) com estes dados. Na prática estas ações (*actions*) são métodos que, depois de executada sua função, irão chamar uma *view* passando os dados manipulados pelo controlador.

O Yii 2.1 utiliza a `yii\web\Controller` como classe base dos controllers (controladores), de maneira semelhante à `CWebController` no Yii 1.1. A `yii\base\Action` é a classe base para classes de actions (ações).

O impacto mais óbvio destas mudanças em seu código é que uma action de um controller deve sempre retornar o conteúdo que quer renderizar ao invés de dar echo nele:

```
public function actionView($id)
{
    $model = Post::findOne($id);
    if ($model) {
        return $this->render('view', ['model' => $model]);
    } else {
        throw new \yii\web\NotFoundException;
    }
}
```

2.4.1.5. Camada de visualização (View)

Esta é a camada responsável por exibir os dados para o usuário final, ela possui comunicação direta com o controlador. As visualizações não possuem nenhuma ligação direta com a camada de Modelos.

A mudança mais significativa das views no Yii 2 é que a variável especial `$this` em uma view não se refere mais ao controle (controlador) ou widget atual. Invés, `$this` agora se refere a um objeto view, um novo conceito introduzido no 2.1. O objeto view é do tipo `yii\web\View`, que representa a parte da visão do padrão MVC. Se quiser associar o controlador (controle) ou o widget em uma visão, pode-se utilizar `$this->context`.

Para renderizar uma visão parcial (partial view) dentro de outra view, usa `$this->render()`, e não `$this->renderPartial()`. A Chamada de render também precisa ser explicitamente imprimida com echo, uma vez que o método `render()` retorna o resultado da renderização ao invés de exibi-lo diretamente. Por exemplo:

```
echo $this->render('_item', ['item' => $item]);
```

2.4.1.6. Widgets

Yii 2.1 usa `yii\base\Widget` como a classe base dos widgets, de maneira semelhante à `CWidget` no Yii 1.1. Para obter um melhor suporte ao framework nas IDEs, o Yii 2.1 introduz uma nova sintaxe para utilização de widgets. Os métodos estáticos `begin()`, `end()` e `widget()` foram introduzidos, para serem utilizados do seguinte modo:

```
use yii\widgets\Menu;

use yii\widgets\ActiveForm;
```

```

// Perceba que você tem que dar um "echo" no resultado para exibi-lo

echo Menu::widget(['items' => $items]);

// Passando um array para inicializar as propriedades do objeto

$form = ActiveForm::begin([

    'options' => ['class' => 'form-horizontal'], 'fieldConfig' => ['inputOptions' =>
['class' => 'input-xlarge']],

    ... campos de entrada do form aqui ...

ActiveForm::end();

```

Os widgets são blocos de construção reutilizáveis usados nas views (visões) para criar e configurar complexos elementos de interface do usuário sob uma modelagem orientada a objetos. Por exemplo, um widget `datepicker` pode gerar um calendário que permite aos usuários selecionarem uma data que desejam inserir em um formulário. Basta inserir um código na view (visão) conforme o seguinte:

```

<?php

use yii\jquery\DatePicker;

?>

<?= DatePicker::widget(['name' => 'date']) ?>

```

Existe uma quantidade considerável de widgets empacotados no Yii, como o active form, o menu, o jQuery UI widgets, o Twitter Bootstrap widgets, etc.

2.4.1.7. Assets

O Yii 2.1 introduz um novo conceito chamado de asset bundle (pacote de assets) que substitui o conceito de script packages (pacotes de script) encontrado no Yii 1.1.

Um asset bundle é uma coleção de arquivos de assets (por exemplo, arquivos JavaScript, arquivos CSS, arquivos de imagens, etc.) dentro de um diretório. Cada asset bundle é

representado por uma classe que estende `yii\web\AssetBundle`. Ao registrar um asset bundle via `yii\web\AssetBundle::register()`, torna os assets destes pacotes acessíveis via Web. Ao contrário do Yii 1, a página que registra o bundle automaticamente conterá as referências aos arquivos JavaScript e CSS especificados naquele bundle.

2.4.1.8. Forms

O Yii 2.1 introduz o conceito de campos (fields) para a construção de formulários usando `yii\widgets\ActiveForm`. Um field é um container consistindo de um label, um input, uma mensagem de erro, e/ou um texto de ajuda. Um field é representado como um objeto `ActiveField`. Usando fields pode construir um formulário de maneira mais limpa do que antes:

```
<?php $form = yii\widgets\ActiveForm::begin(); ?>
    <?= $form->field($model, 'username') ?>
    <?= $form->field($model, 'password')->passwordInput() ?>
    <div class="form-group">
        <?= Html::submitButton('Login') ?>
    </div>
<?php yii\widgets\ActiveForm::end(); ?>
```

2.4.1.9. MCV no PHP

Diferentemente de outras linguagens, como o C# através da plataforma .NET, o PHP possui algumas “dificuldades” para se adaptar ao MVC. “Dificuldades”, pois, comparado às aplicações em C# que só rodam utilizando o framework .NET, o PHP não possui nenhum framework base para rodar suas aplicações. Além disso, há o fato de o PHP ser interpretado, e não compilado, como nas linguagens que utilizam o framework .NET.

Por isso, para se adequar ao **MVC** no **PHP**, é necessário criar suas estruturas de diretórios manualmente, além de criar adaptações para haver a interação entre as três principais camadas do MVC. O Yii implementa o padrão de desenvolvimento modelo-visão-control (MVC) que é amplamente adotado na programação Web. O MVC visa separar a lógica de negócio da interface com o usuário, assim os programadores podem mudar facilmente cada parte, sem afetar as outras. No padrão MVC, o modelo representa as

informações (os dados) e as regras de negócio, a visão contém elemento de interface com o usuário, como textos, formulários, e o controle gere a comunicação entre o modelo e a visão.

Além MVC, o Yii também introduz um controle de frente, chamada aplicação (*application*), que representa o contexto de execução dos processos requisitados. A aplicação recebe a solicitação do usuário e a envia para um controlador adequado para ser processada.

O diagrama seguinte mostra a estrutura estática de uma aplicação Yii:

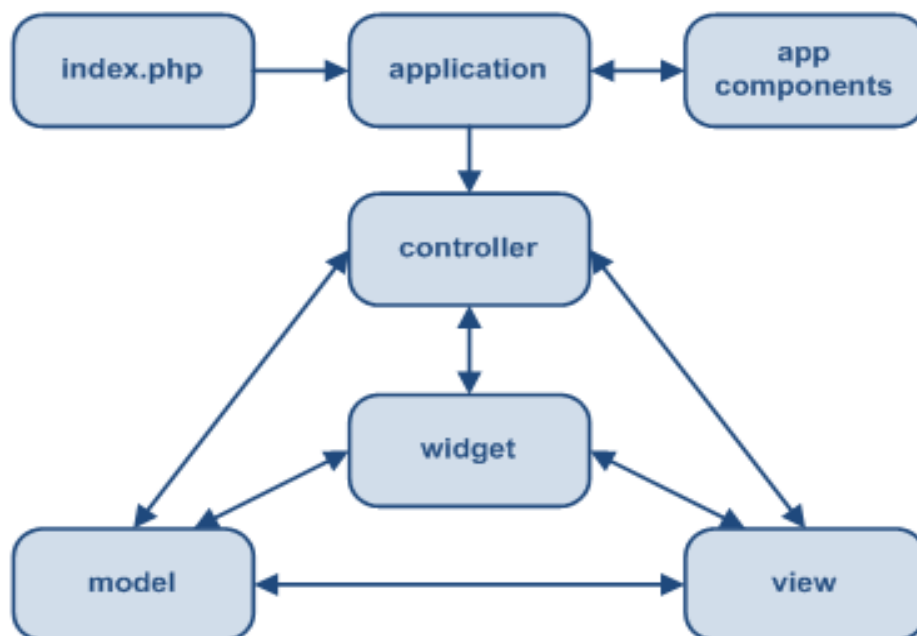


Figura 3: Diagrama da Estrutura Estática de App no Yii

Fonte: <https://www.yiiframework.com/doc/guide/1.1/pt/basics.mvc>

2.4.1.10. AdminLTE

O AdminLTE é um modelo WebApp de software livre popular para painéis de administração e painéis de controle. É um modelo HTML responsivo que é baseado no framework CSS Bootstrap 3. Ele utiliza todos os componentes do Bootstrap em seu design e redesenha muitos plugins comumente usados para criar um design consistente que pode ser usado como uma interface de usuário para aplicativos de backend.

O AdminLTE é baseado em um design modular, que permite que ele seja facilmente customizado e construído. yii2-adminlte é um pacote para o yii framework 2 que consiste em um modelo de backend muito popular. Este pacote inclui requisitos de temas e widgets de base para uso confortável em seus aplicativos da web.

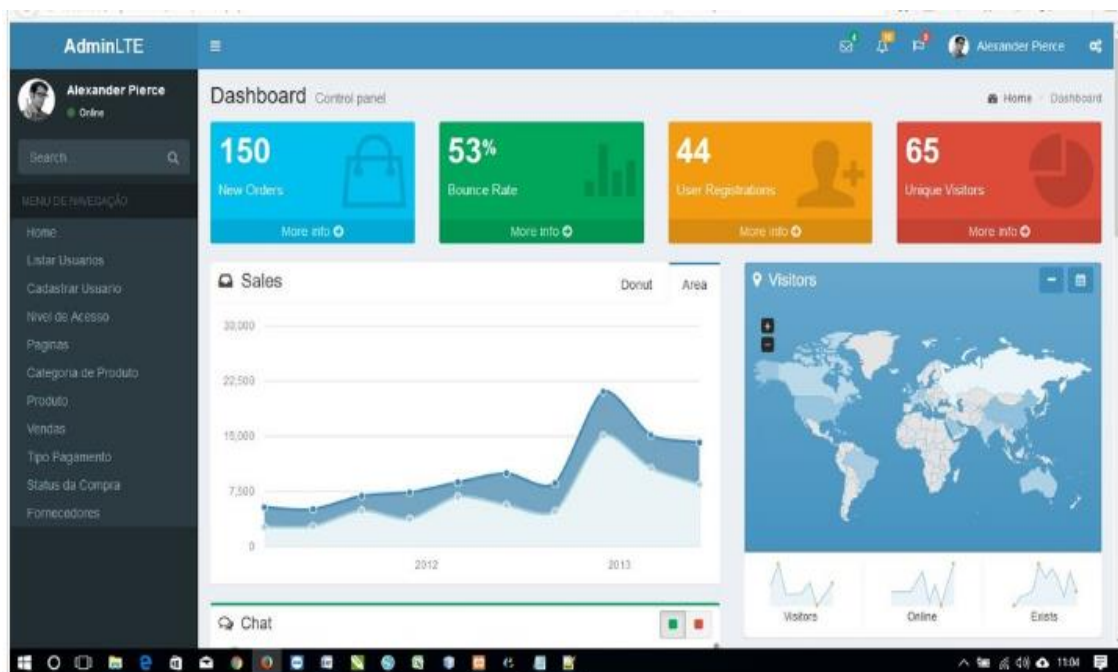


Figura 4: Tela do AdminLTE

Fonte: <https://adminlte.io/themes/AdminLTE/index.html>

yii2-adminlte-asset Pacote de recursos AdminLTE para o tema Backend. Interface de usuário de back-end para o Yii2 Framework, com base no AdminLTE. Este pacote contém um pacote de ativos para o Yii 2 Framework, que registra os arquivos CSS para a interface de usuário AdminLTE. Os arquivos CSS são instalados através do uso recomendado pelo Yii do `fxp/composer-asset-plugin`

2.4.2. PHP

É uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na *World Wide Web*. Figura entre as primeiras linguagens possíveis de inserção em documentos HTML, dispensando em muitos casos o uso de arquivos externos para

eventuais processamentos de dados. O código é interpretado no lado do servidor pelo módulo PHP, que também gera a página web a ser visualizada no lado do cliente. A linguagem evoluiu, passou a oferecer funcionalidades em linha de comando, e além disso, ganhou características adicionais, que possibilitaram usos adicionais do PHP, não relacionados a *web sites*.

De acordo com NIEDERAUER (2007) “PHP é uma das linguagens de programação mais utilizadas na web para a criação de páginas dinâmicas.

Principais características da linguagem de programação PHP:

- Gratuito e com código aberto: o arquivo de instalação pode ser adquirido gratuitamente no site <http://www.php.net>. Além disso, o PHP é um software com código-fonte aberto;
- Embutido no HTML: o HTML e o PHP estão misturados. Você pode começar a escrever em PHP, de repente escrever um trecho em HTML, depois voltar para o PHP, e assim por diante;
- Baseado no servidor: quando acedemos a uma página PHP através do seu navegador, todo o código PHP é executado no servidor, e somente o resultado final é exibido para o utilizador. Portanto, o navegador exibe a página já processada, sem consumir recursos de seu computador;
- Bancos de dados: diversos bancos de dados são suportados pelo PHP, ou seja, o PHP possui código que executa funções de cada um. Entre eles temos PostgreSQL, Sybase, MySQL, Oracle, SQL Server e muitos outros;
- Portabilidade: pode-se executar o PHP no Linux, Unix ou Windows.”

Vantagens da linguagem PHP

O PHP possui algumas vantagens tais como:

- Fácil aprendizagem – A linguagem PHP adquiriu elementos do Perl, Java e do C. A maioria dos programadores para Web conhece ao menos uma destas linguagens, facilitando o aprendizado;
- Acesso a dados – O PHP se conecta facilmente a sistemas Sybase, MySQL, MS-SQL, Oracle e muitos outros compatíveis com o padrão ODBC;

- Velocidade e robustez – O ASP tende a deixar o servidor mais lento, o PHP raramente causa este problema. A diferença entre ambos já foi maior, mas o ASP foi sendo otimizado pela Microsoft enquanto o PHP cresceu, aglutinando mais e mais funções;
- Multiplataforma – o PHP funciona em qualquer plataforma onde for possível instalar um servidor Web.

Desvantagens do PHP

Essa linguagem apresenta alguns pontos fracos tais como:

- Compatibilidade entre versões – Nota-se que falta um pouco de padronização. Por exemplo, um comando que funciona em determinada revisão pode não funcionar em outra;
- Documentação incompleta – Como a maioria dos programas Open Source, frequentemente os recursos surgem antes de estarem documentados. O site da organização ajuda bastante, mas é comum encontrar recursos sem documentação e, principalmente, sem exemplos que possam facilitar a aprendizagem. Mas é preciso dizer que os recursos sem documentação só serão importantes para quem é utilizador bem avançado, ou seja, ele já saberá o que fazer para contornar a falta de documentação.

2.4.3. HTML

É a sigla de **HyperText Markup Language**, expressão inglesa que significa "Linguagem de Marcação de Hipertexto". Consiste em uma linguagem de marcação utilizada para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet.

Para escrever documentos HTML não é necessário mais do que um editor de texto simples e conhecimento dos códigos que compõem a linguagem. Os códigos (conhecidos como tags) servem para indicar a função de cada elemento da página Web. Os tags funcionam como comandos de formatação de textos, formulários, links (ligações), imagens, tabelas, entre outros.

Os navegadores (browsers) identificam as *tags* e apresentam a página conforme está especificada. Um documento em HTML é um texto simples, que pode ser editado no Bloco de Notas (Windows) ou Editor de Texto (Mac) e transformado em hipertexto. A linguagem HTML foi criada por Tim Barners Lee na década de 1991. As especificações da linguagem são controladas pela W3C (World Wide Web Consortium).

2.4.4. JQuery

JQuery é uma biblioteca de código aberto e possui licença dual, fazendo uso da Licença MIT ou da GNU General Public License versão 2. A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos e desenvolver aplicações AJAX.

A biblioteca também oferece a possibilidade de criação de plugins sobre ela. Fazendo uso de tais facilidades, os desenvolvedores podem criar camadas de abstração para interações de mais baixo nível, simplificando o desenvolvimento de aplicações web dinâmicas de grande complexidade.

2.4.5. CSS

Cascading Style Sheets (CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. O seu principal benefício é a separação entre o formato e o conteúdo de um documento. Em vez de colocar a formatação dentro do documento, o desenvolvedor cria uma ligação (Link) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta modificar apenas um arquivo.

Com a variação de atualizações dos navegadores (browsers) como Internet Explorer que ficou sem nova versão de 2001 a 2006, o suporte ao CSS pode variar. O Internet Explorer 6, por exemplo, tem suporte total a CSS1 e praticamente nulo a CSS2. Navegadores mais modernos como Google Chrome e Mozilla Firefox tem suporte maior, inclusive até a CSS3, ainda em desenvolvimento.

2.4.6. Base de Dados MySQL

O MySQL é um sistema de gestão de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos bancos de dados mais populares, com mais de 11 milhões de instalações pelo mundo.

Entre os usuários do banco de dados MySQL estão: NASA, Friendster, Banco Bradesco, Dataprev, HP, Nokia, Sony, Lufthansa, U.S Army, US. Federal Reserve Bank, Associated Press, Alcatel, Slashdot, Cisco Systems, Google, CanaVialis S.A e outros.

Criado na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que têm trabalhado juntos desde a década de 1981. Hoje seu desenvolvimento e manutenção empregam aproximadamente 411 profissionais no mundo inteiro, e mais de mil contribuem testando o *software*, integrando-o a outros produtos. O sucesso do MySQL deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de hospedagem de *sites* da Internet oferecidos atualmente.

2.4.7. MySQL Workbench

O MySQL Workbench é uma ferramenta com interface gráfica que permite modelar base de dados. Foi criado pela MySQL AB e é sem dúvida uma ferramenta muito completa e útil para quem trabalha com base de dados.

Principais funcionalidades:

- Criação de diagrama EER;
- SQL Scripts;
- Catálogo da Base de Dados;
- Visualização total do *Dashboard*;
- Informações sobre o objeto selecionado;
- Gestão de ligações a Servidores MySQL;
- *Forward Engineering* e Engenharia Reversa.

2.4.8. UML

Linguagem de Modelagem Unificada (do inglês, **UML - Unified Modeling Language**) é uma linguagem de modelagem que permite representar um sistema de forma padronizada.

A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz o que fazer primeiro e em seguida ou como projetar um sistema, mas ela auxilia a visualizar o desenho e a comunicação entre os objetos. Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica.

É uma notação independente de processos, embora o RUP (*Rational Unified Process*) tenha sido especificamente desenvolvido utilizando a UML. É importante distinguir entre um modelo UML e um diagrama (ou conjunto de diagramas) de UML. O último é uma representação gráfica da informação do primeiro, mas o primeiro pode existir independentemente. O XMI (*XML Metadata Interchange*) na sua versão corrente disponibiliza troca de modelos, mas não de diagramas.

Os objetivos da UML são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.

2.4.9. Apache

O **servidor Apache** (ou Servidor HTTP Apache, em inglês: *Apache HTTP Server*, ou simplesmente: **Apache**) é o mais bem-sucedido servidor web livre. Foi criado em 1995 por Rob McCool.

Em uma pesquisa realizada em dezembro de 2007, foi constatado que a utilização do Apache representa cerca de 47.21% dos servidores ativos no mundo. Em maio de 2001, o Apache serviu aproximadamente 54,68% de todos os *sites* e mais de 66% dos milhões de sites mais movimentados. É a principal tecnologia da *Apache Software Foundation*,

responsável por mais de uma dezena de projetos envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos.

O servidor é compatível com o protocolo HTTP versão 1.1. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos — utilizando a API do software.

2.4.10. Bootstrap

A Bootstrap é uma framework front-end criada pelos desenvolvedores do Twitter para agilizar desenvolvimento de sítios web e aplicações web. Ela é um conjunto de recursos visuais (css, less, ícones), estruturais (grids, navegação e html semântico) e dinâmicos (javascript e ajax).

Bootstrap foi desenvolvido por Mark Otto e Jacob Thornton no Twitter, e lançado como um produto de código aberto em agosto de 2001 no GitHub. Bootstrap é uma estrutura de front-end livre para desenvolvimento web mais rápida e fácil, inclui HTML e CSS modelos de design com base para tipografia, formas, botões, tabelas, navegação, modais, carrosséis de imagem e muitos outros, bem como plugins JavaScript opcionais.

Bootstrap também lhe dá a capacidade de facilmente criar projetos. Uma característica interessante da Bootstrap é que ela usa Less (uma biblioteca que adiciona ao CSS recursos de linguagem de programação como variáveis e funções). Isso permite definir padrões para serem utilizados em um sítio web inteiro, além de automatizar algumas tarefas repetitivas. Também permite modificar o visual da framework e criar temas personalizados.

Alguns sítios web disponibilizam temas para download de graça ou por valores simbólicos, além de outros componentes para estender as funcionalidades ou acrescentar novas, tornando Bootstrap ainda mais poderosa.

2.4.11. JavaScript

JavaScript foi inventado por Brendan Eich, em 1995, e tornou-se um padrão ECMA em 1997. JavaScript é a linguagem de programação mais popular do mundo. JavaScript é

uma linguagem interpretada. Isto significa que o código é executado diretamente (sem compilação prévia) quando a página HTML é carregada no browser. O JavaScript é o hoje suportado por praticamente todas as plataformas WWW - World Wide Web na internet.

Tendo aparecido como linguagem a executar em ambiente browser, o JavaScript é totalmente baseado no ambiente da WWW. Assim, os programas JavaScript funcionarão associados às ocorrências (eventos) que sejam desencadeadas numa determinada página. O JavaScript permite inserir vários efeitos, fazendo com que a sua página Web fique mais dinâmico.

Características do JavaScript

- Controlar o comportamento do navegador: como criar janelas pop-up, apresentar mensagens ao utilizador, alterar as dimensões do navegador, interferir na barra de estados, retirar menus, fechar e abrir janelas, entre outras;
- É a linguagem de programação do lado do cliente, porque é o navegador que suporta a carga de processamento;
- Permite inserir vários efeitos, fazendo com que o site fique mais dinâmico: em conjunto com o HTML, além de facilitar a validação de formulários, o JavaScript permite inserir vários efeitos, fazendo com que o site (ou uma mensagem de correio eletrônico) fique mais dinâmico, e com maior interação com o utilizador;
- O código JavaScript deve ser colocado junto aos comandos da linguagem HTML, Como o código *JavaScript* não é compilado, ele deve ser colocado no código fonte, junto aos comandos da linguagem HTML, podendo essa ação ser realizado em arquivo à parte;
- O JavaScript coloca à disposição do programador tudo que forma a página Web, para que esta possa acessá-los e modificá-los dinamicamente.

CAPITULO III

3. Modelação do Sistema

Neste capítulo são apresentadas as características do sistema desenvolvido. Ainda são, descritas as especificações dos requisitos funcionais e não funcionais, dos diagramas de casos de uso, de classes e de sequencia, e para as respectivas modelagem foi utilizada MySQLWorkBench e o Visual Paradigm.

3.1. Análises de Requisitos do Sistema

Requisito funcional

Define uma função de um software ou parte dele. Ele é o conjunto de entradas. O seu comportamento e a sua saída, ou seja, envolvem cálculos, lógicas de trabalho, manipulação e processamento e dados, entre outros. Dentro dos requisitos funcionais também se encontram a arquitetura do aplicativo, diferentemente da arquitetura técnica, que pertence aos requisitos não funcionais.

Requisitos não funcionais

São relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, disponibilidade, segurança e tecnologia envolvidas. Muitas vezes, os requisitos não funcionais acabam gerando restrições aos funcionais. Entender profundamente os tipos de requisitos no momento de definir o seu software pode decidir o futuro dele. Por outro lado, também é preciso ter em mente que o software pode ser beneficiado se houver um pouco de flexibilidade no seu desenvolvimento. Assim, unindo os conceitos, é possível ter um sistema de qualidade sem um alto custo de aquisição.

3.1.1. Requisitos funcionais

Este requisito define uma função de um sistema de software ou seu componente. Uma função é descrita como um conjunto de entradas, seu comportamento e as saídas.

Para esta aplicação foram definidas os seguintes requisitos funcionais:

RF1: O sistema deverá permitir agendar consultas;

RF2: O sistema deverá permitir ao paciente visualizar, agendar suas consultas;

RF3: O sistema deverá permitir acesso a informações mediante autenticação do usuário;

RF4: O sistema deverá permitir ao médico visualizar resultados;

RF5: O sistema deverá permitir ao funcionário cadastrar os pacientes;

RF6: O sistema deverá permitir ao médico, aos funcionários e pacientes que cancelem um agendamento;

RF7: O sistema deverá permitir aos pacientes visualizar os horários disponíveis para o agendamento;

RF8: O sistema deverá permitir o médico e seus funcionários visualizar uma agenda que relaciona todas as consultas agendadas e os horários disponíveis.

3.1.2. Requisitos não funcionais

RNF1: O sistema deverá utilizar banco de dados MySQL;

RNF2: Será utilizado Apache HTTPD para disponibilizar o site na rede;

RNF3: O sistema será desenvolvido na linguagem PHP;

RNF4: A aplicação deve funcionar com qualquer tipo de browser.

3.2. Desenho do Sistema

Para fazer a modelação do Sistema foi utilizado uma notação gráfica standard - UML - que modela o sistema independentemente da linguagem de programação.

A UML disponibiliza um conjunto de diferentes tipos de diagramas, dentre as quais serão estudados:

- Diagrama de Use Cases;
- Diagrama de Classes;
- Diagrama de Sequência;
- Diagrama de Actividade;

3.2.1. Diagramas de caso de uso

Segundo Alberto Silva e Carlos Videira (2001), “um diagrama de casos de utilização descreve a relação entre atores e casos de utilização de um dado sistema permitindo dar uma visão global e de alto nível do sistema.

Casos de uso são tipicamente relacionados a "atores". Um ator é um humano ou entidade máquina que interage com o sistema para executar um significativo trabalho”.

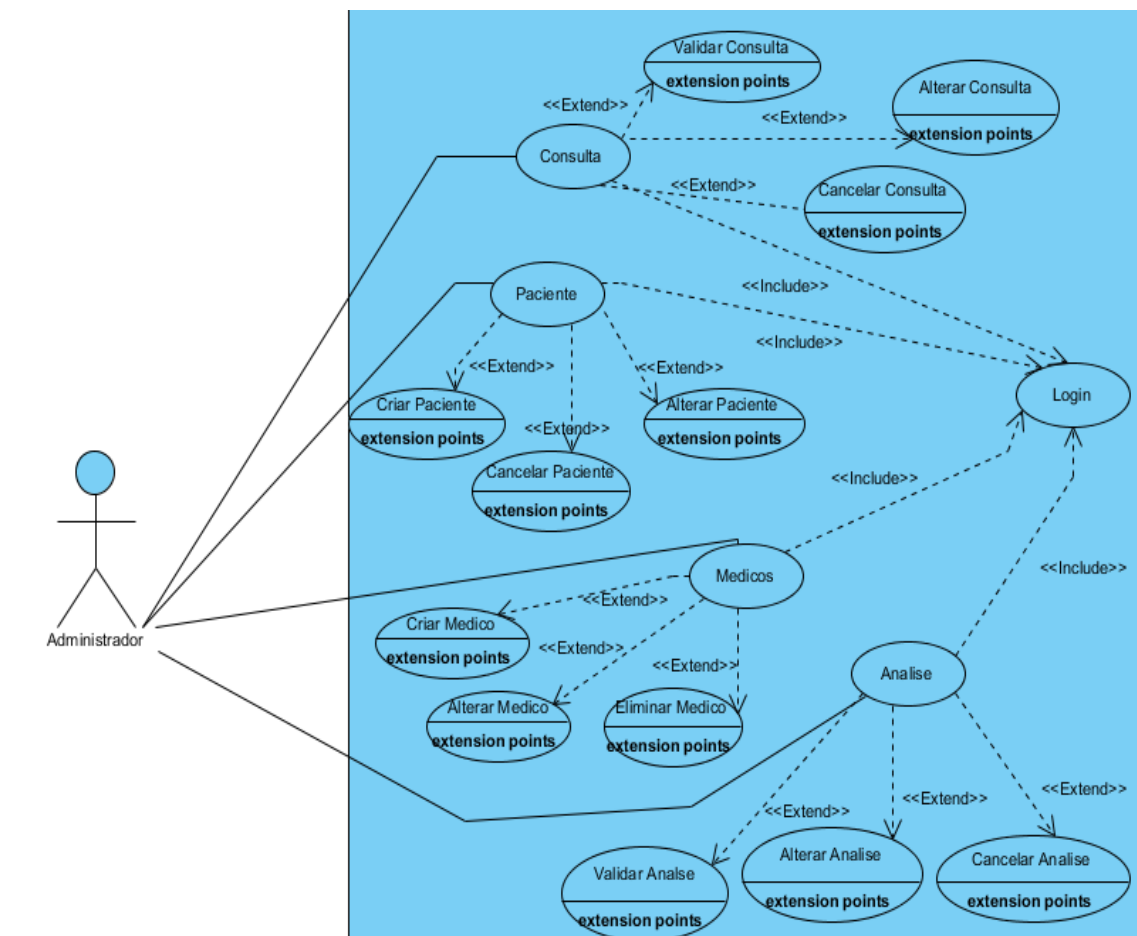


Figura 5: Diagrama caso uso Administrador

Fonte: Elaborado pela autora desse trabalho

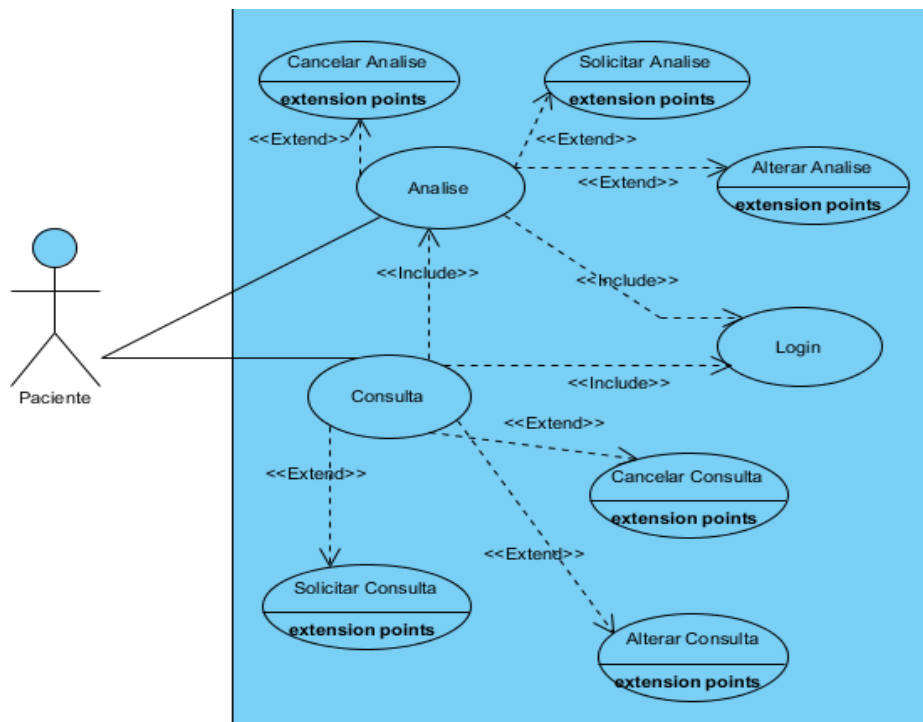


Figura 6: Diagrama caso uso do Paciente

Fonte: Elaborado pela autora desse trabalho

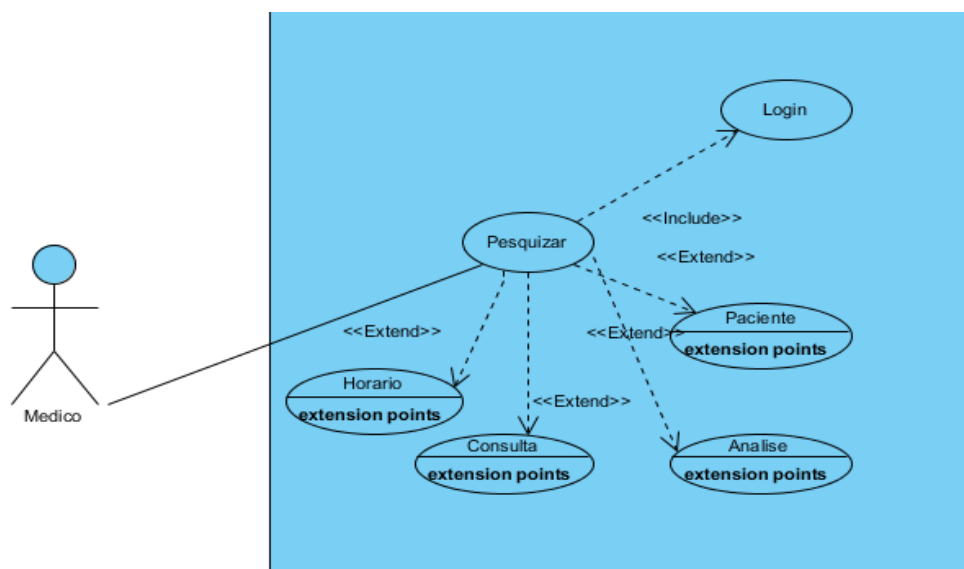


Figura 7: Diagrama caso uso do Medico

Fonte: Elaborado pela autora desse trabalho

3.2.2. Diagrama de classes

Os diagramas de classes descrevem a estrutura estática de um sistema, em particular as entidades existentes, as suas estruturas internas, e relações entre si. Permitem também ilustrar os detalhes de um sistema em determinado momento ao providenciarem cenários de possíveis configurações.

De acordo com SILVA e VIDEIRA (2001), “uma classe é a descrição de um conjunto de objetos que partilham os mesmos atributos, operações, relações e a mesma semântica. Uma classe corresponde a algo tangível ou a uma abstração conceptual existente no domínio do utilizador ou no domínio do engenheiro de *software*.”

Segundo os mesmos, uma classe bem estruturada.

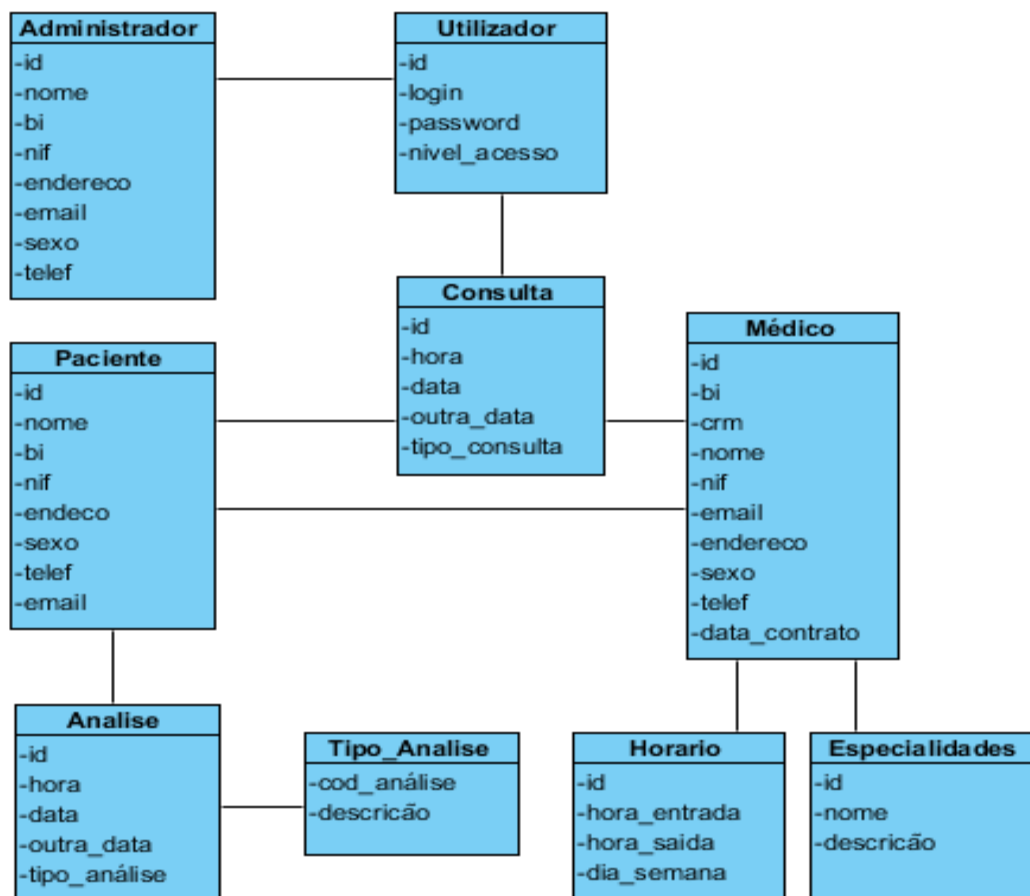


Figura 8: Diagrama de Classe do sistema

Fonte: Elaborado pela autora desse trabalho

3.2.3. Diagrama de sequencia

Caso de Uso: Fazer Login.

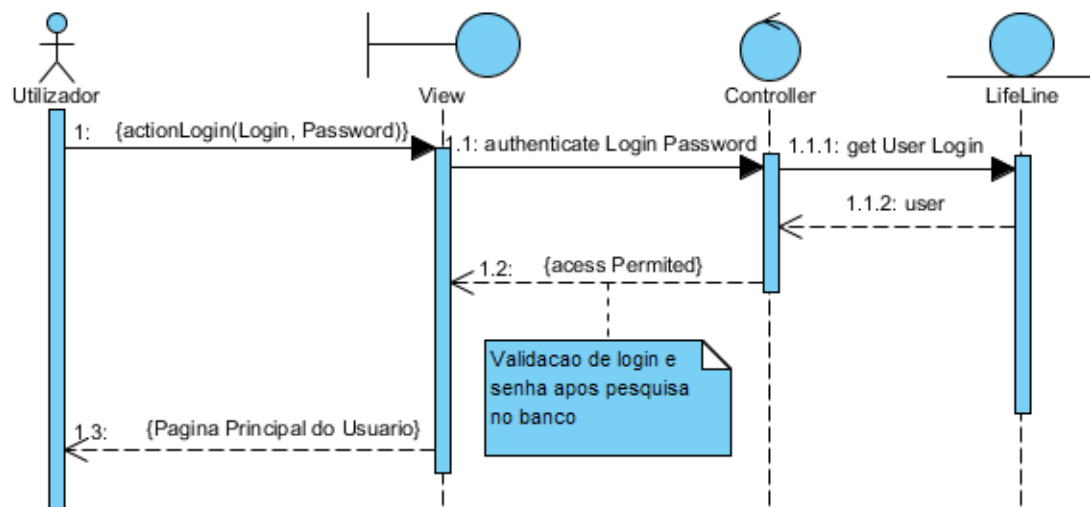


Figura 9: Diagrama sequencia Fazer Login

Fonte: Elaborado pela autora desse trabalho

Caso de Uso: Registrar Paciente.

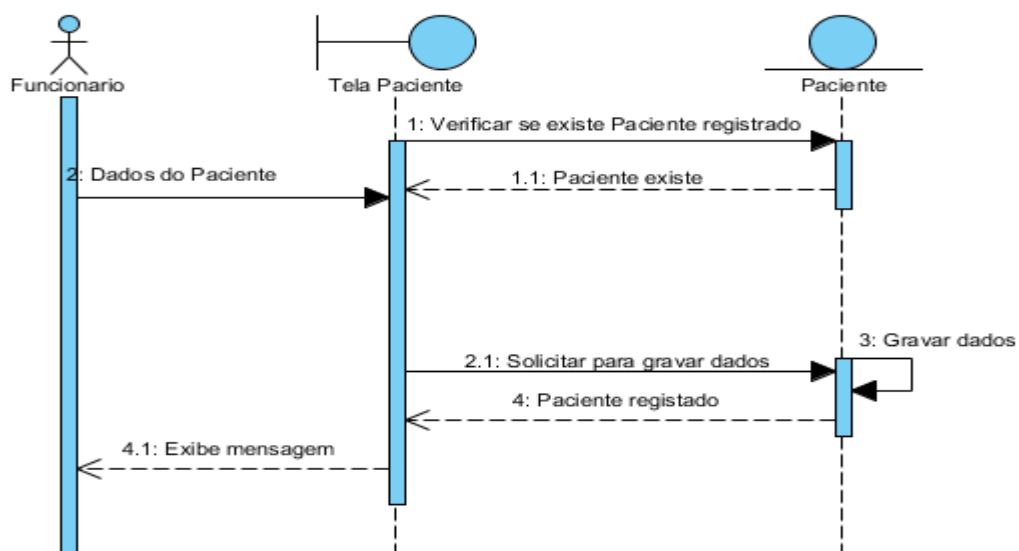


Figura 10: Diagrama sequencia Registrar Paciente

Fonte: Elaborado pela autora desse trabalho

3.2.4. Dicionário de dados

O Dicionário de dados (DD) é de fundamental importância na modelagem dos requisitos do utilizador. Ele fornece uma listagem organizada com definições precisas e rigorosas (quase formais) de todos os elementos de dados pertinentes ao sistema. Assim, devem ser definidos no Dicionário de Dados todos os fluxos de dados e componentes de depósito de dados.

Além da descrição dos dados do sistema, o dicionário, em geral, inclui a descrição das entidades externas e dos processos no Diagrama de Fluxo de Dados. A descrição do procedimento (política) de processo é importante para descrever o comportamento dos processos. A descrição dos processos primitivos neste é particularmente importante, visto que para esses processos não são desenhados DFD's de nível mais detalhado.

Tabelas que descrevem o Dicionário de Dados:

Análise – Entidade que define a análise.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Hora	time	
Data	date	
estado	varchar	45
quantidade	double	
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 2: Dicionário de dado da tabela Analise

Consulta – Entidade que define a consulta.

Atributo	Tipos de Dados	Tamanho
Id	Varchar	36
Hora	Time	
Data	Date	
estado	Varchar	1
descrição	Varchar	311
date_create	Datetime	
date_update	Datetime	
user_create	Varchar	36
user_update	Varchar	36

Tabela 3: Dicionário de dado da tabela Consulta

Boletim_Clínico– Entidade que define o boletim do paciente.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
sintomas	varchar	45
avaliação	varchar	45
hipótese_diagnostico	varchar	45
tratamento	varchar	45
orientações	varchar	45
estado	varchar	1
Observações	text	
grupo sanguíneo	varchar	11
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 4: Dicionário de dado da tabela Bolitim_Clínico

Paciente – Entidade que define o paciente.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	311
username	varchar	61
Password	varchar	111
auth_key	varchar	32
Password_reset_token	varchar	111
estado	varchar	1
date_insert	datetime	

Tabela 5: Dicionário de dado da tabela Paciente

Receita – Entidade que define a receita paciente.

Atributo	Tipos de Dados	Tamanho
Id	Varchar	36
Data	Date	
estado	Varchar	1
observação	Text	

Tabela 6: Dicionário de dado da tabela Receita

Calendário – Entidade que define o calendário.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Data	date	
título	varchar	111
descrição	varchar	2111
estado	varchar	1

Tabela 7: Dicionário de dado da tabela Calendário

Complemento – Entidade que define o complemento.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
documento	varchar	45
num_documento	int	11
nif	int	11
sexo	varchar	36
data_nascimento	date	
telefone	varchar	45
telemóvel	varchar	45
filiação	varchar	311
endereço	varchar	36
estado_civil	varchar	36
email	varchar	45
estado	varchar	1

Tabela 8: Dicionário de dado da tabela Complemento

Contato– Entidade que define o contato.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Nome assunto	varchar	111
email	varchar	45
cidade	int	11
assunto	int	11
mensagem	text	

Tabela 9: Dicionário de dado da tabela Contato

Domínio – Entidade que define o domínio.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
domínio	varchar	21
significado	varchar	151
valor	varchar	11
estado	varchar	1

Tabela 10: Dicionário de dado da tabela Domínio

Empresa – Entidade que define a empresa.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	311
Tipo_sociedade	varchar	1
Nif	int	11
cidade	varchar	45
concelho	varchar	45
Pais	varchar	45
telefone	varchar	45
telemóvel	varchar	45
website	varchar	45
Fax	varchar	45
email	varchar	45
estado	varchar	1

Tabela 11: Dicionário de dado da tabela Empresa

Especialidade– Entidade que define a especialidade.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	311
estado	varchar	1

Tabela 12: Dicionário de dado da tabela Especialidade

Geografia– Entidade que define a geografia.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
zona	varchar	51
conselho	varchar	51
estado	varchar	1
pais_codigo	int	11

Tabela 13: Dicionário de dado da tabela Geografia

Ilhas – Entidade que define a ilhas.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	35
código	varchar	11
estado	varchar	1

Tabela 14: Dicionário de dado da tabela Ilhas

Login – Entidade que define a login.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	311
username	varchar	61
Password	varchar	111
auth_key	varchar	32
Password_reset_token	varchar	111
estado	varchar	1
date_insert	datetime	
endereço	varchar	45
telefone	varchar	45
telemóvel	varchar	45
email	varchar	45
perfil_id	varchar	36
domínio_id	varchar	36

Tabela 15: Dicionário de dado da tabela Login

LoginValidation – Entidade que define a loginvalidation.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
token	varchar	111
data_insert	datetime	
data_validation	datetime	
login_id	varchar	36
estado	varchar	1

Tabela 16: Dicionário de dado da tabela LoginValidation

Medicamento – Entidade que define a medicamento.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	111
descrição	varchar	311
dosagem	varchar	45
estado	varchar	1
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 17: Dicionário de dado da tabela Medicamento

Medicamento_Receita – Entidade que define a medicamento_receita.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
quantidade	int	11
dosagem	text	
medicamento_id	varchar	36
receita_id	varchar	36

Tabela 18: Dicionário de dado da tabela Medicamento_Receita

Medico_Especialidade – Entidade que define a medico_especialidade.

Atributo	Tipos de Dados	Tamanho
login_id	varchar	36
especialidade_id	varchar	36

Tabela 19: Dicionário de dado da tabela Medico_Especialidade

Menu – Entidade que define a menu.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Menu	varchar	111
Icon	varchar	15
url	varchar	51
Estado	varchar	1
Contexto	varchar	1
Ordem	int	11
menu_id	int	11

Tabela 20: Dicionário de dado da tabela Menu

MenuPerfil – Entidade que define a menuperfil.

Atributo	Tipos de Dados	Tamanho
Id	int	11
Estado	varchar	1
menu_id	int	11
perfil_id	varchar	36

Tabela 21: Dicionário de dado da tabela MenuPerfil

Pais – Entidade que define a pais.

Atributo	Tipos de Dados	Tamanho
codigo	int	11
fone	int	11
Isso	varchar	41
Iso3	varchar	41
nome	varchar	111
nomeFormal	varchar	111
estado	varchar	1

Tabela 22: Dicionário de dado da tabela Pais

Perfil – Entidade que define a perfil.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
designação	varchar	36
estado	varchar	1

Tabela 23: Dicionário de dado da tabela Perfil

Resultado_Analise – Entidade que define a resultado_analise.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
estado	varchar	1
status	varchar	36
data_colheita	date_time	
código_colheita	varchar	111
data resultado	date	
hipótese_diagnostico	text	
resultado	text	
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36
analise_id	varchar	36
paciente_id	varchar	36
consulta_id	varchar	36
medico_id	varchar	36

Tabela 24: Dicionário de dado da tabela Resultado_Analise

Especialidades – Entidade que define a especialidade, esta tabela é referente ao site.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Título	varchar	51
Descrição	varchar	2111
leia_mais	varchar	2111
Estado	varchar	1

Tabela 25: Dicionário de dado da tabela Especialidades

Tipo_Analise – Entidade que define a tipo_analise.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Nome	varchar	51
estado	varchar	1
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 26: Dicionário de dado da tabela Tipo_Analise

Horario_Consulta – Entidade que define a horario_consulta, esta tabela é referente ao site.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
dia_semana	varchar	36
Primeiro_periodo	varchar	51
segundo_periodo	varchar	51
Estado	varchar	1

Tabela 27: Dicionário de dado da tabela Horario_Consulta

Tratamento – Entidade que define a tratamento.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Procedimento	varchar	45
Observação	varchar	45
Estado	varchar	1
paciente_id	varchar	36
consulta_id	varchar	36
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 28: Dicionário de dado da tabela Tratamento

Documentos_Pacientes – Entidade que define a documentos_pacientes.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
nome	varchar	111
anexo	varchar	45
paciente_id	varchar	36
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36

Tabela 29: Dicionário de dado da tabela Documentos_Pacientes

Dados_Clínicos – Entidade que define a dados_clínicos.

Atributo	Tipos de Dados	Tamanho
Id	varchar	36
Grupo_sanguineo	varchar	45
altura	varchar	45
peso	varchar	45
histórico	varchar	45
paciente_id	varchar	36
date_create	datetime	
date_update	datetime	
user_create	varchar	36
user_update	varchar	36
estado	varchar	1

Tabela 30: Dicionário de dado da tabela Dados_Clínicos

Migration – Entidade que define migration.

Atributo	Tipos de Dados	Tamanho
version	varchar	181
apply_time	int	11

Tabela 31: Dicionário de dado da tabela Migration

Estes quatros últimas tabelas são as que fazem parte das Permissões de base de dados.

Auth_Assignment – Entidade que define auth_assignment.

Atributo	Tipos de Dados	Tamanho
item_name	varchar	64
user_id	varchar	64
created_at	int	11

Tabela 32: Dicionário de dado da tabela Auth_Assignment

Auth_Item – Entidade que define auth_item.

Atributo	Tipos de Dados	Tamanho
name	varchar	64
Type	smallint	6
description	text	
rule_name	varchar	64
Data	blob	
Created_at	int	11
Updated_at	int	11

Tabela 33: Dicionário de dado da tabela Auth_Item

Auth_Item _Child– Entidade que define auth_item _child.

Atributo	Tipos de Dados	Tamanho
parent	varchar	64
child	varchar	64

Tabela 34: Dicionário de dado da tabela Auth_Item _Child

Auth_Rule – Entidade que define auth_rule.

Atributo	Tipos de Dados	Tamanho
Name	varchar	64
Data	blob	
Created_at	int	11
Updated_at	int	11

Tabela 35: Dicionário de dado da tabela Auth_Rule

CAPITULO IV

4. Implementação do Protótipo

De seguida apresenta-se as fases do desenvolvimento do sistema de marcação de consulta e análise online. Nesta seção são mostradas as técnicas e ferramentas utilizadas, a operacionalidade da implementação, bem como a codificação do sistema.

Para o desenvolvimento do protótipo foram utilizadas várias ferramentas e tecnologias, como as linguagens de programação PHP, HTML, MySQL para o armazenamento de dados, a linguagem modelação UML para a criação dos diagramas, Bootstrap e CSS para o embelezamento das páginas e gestão de conteúdos, porém neste capítulo fez-se um resumo geral de cada um visto que já se encontram descritas mais aprofundadamente no capítulo II no tópico de ferramentas e tecnologias utilizadas.

4.1. Ferramentas e tecnologias

PHP - é a ferramenta que permite a criação das páginas dinâmicas, capaz de ser embebido dentro do código HTML e efetuar determinadas operações capazes de gerar páginas instantaneamente. Esta linguagem foi a escolhida para a implementação do webservice que será o responsável para o tratamento das atualizações e requisições do sistema.

```
1  <?php
2
3  use yii\helpers\Html;
4
5
6  /* @var $this yii\web\View */
7  /* @var $model common\models\Analise */
8
9  $this->title = 'Create Analise';
10 $this->params['breadcrumbs'][] = ['label' => 'Analises', 'url' => ['index']];
11 $this->params['breadcrumbs'][] = $this->title;
12 ?>
13 <div class="analise-create">
14
15     <h1><?= Html::encode($this->title) ?></h1>
16
17     <?= $this->render('_form', [
18         'model' => $model, 'local' => $local, 'tipo_analise' => $tipo_analise,
19     ]) ?>
20
21 </div>
22
```

Figura 11: Exemplo de um Código PHP

HTML - é baseado no conceito de Hipertexto que são conjuntos de elementos ou nós interligados. Estes elementos podem ser palavras, imagens, vídeos, áudio, documentos que quando ligados formam uma grande rede de informação.

Esta linguagem foi escolhida por ser a única linguagem que o navegador consegue interpretar a exibição de conteúdo. Para escrever documentos HTML não é necessário mais do que um editor de texto simples e conhecimento dos códigos que compõem a linguagem. Os códigos servem para indicar a função de cada elemento da página Web e funcionam como comandos de formatação de textos, formulários, links (ligações), imagens, tabelas, entre outros.

```
11 <div class="menu-form">
12     <?php $form = ActiveForm::begin(); ?>
13     <div class="box box-info">
14         <div class="box-header with-border">
15             <h3 class="box-title"><?= Html::encode($this->title) ?></h3>
16         </div><!-- /.box-header -->
17         <div class="panel-body">
18             <div class="row vdivide">
19                 <div class="col-xs-5">
20                     <?= $form->field($model, 'menu')->textInput(['maxlength' => true]) ?>
21                 </div>
22                 <div class="col-xs-2">
23                     <?= $form->field($model, 'icon')->textInput(['maxlength' => true]) ?>
24                 </div>
25                 <div class="col-xs-5">
26                     <?= $form->field($model, 'url')->textInput(['maxlength' => true]) ?>
27                 </div>
28             </div>
29             <div class="row vdivide">
30                 <div class="col-xs-4">
```

Figura 12: Exemplo de um Código HTML

MySQL - É um sistema de gestão de bases de dados relacionais, suporta SQL, é open source e é um dos SGBD para utilização profissional mais utilizado a nível mundial. NEVES, P e RUAS, R, (2005,p.21). Na Base de Dados do sistema desenvolvido estão armazenados as informações referente ao login, pacientes, especialidades, o calendário,

todos os conteúdos do Sistema. As senhas do login estão protegidas através de mecanismos de criptografia.

The image displays two screenshots of the phpMyAdmin interface, showing the database structure of the 'simarc' database. The top screenshot shows tables from 'analise' to 'ilhas'. The bottom screenshot shows tables from 'login' to 'tratamento', with a summary row for 34 tables.

Tabela	Ações	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
analise	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	112 KB	-
auth_assignment	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2	InnoDB	utf8_unicode_ci	32 KB	-
auth_item	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	280	InnoDB	utf8_unicode_ci	48 KB	-
auth_item_child	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	17	InnoDB	utf8_unicode_ci	32 KB	-
auth_rule	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	utf8_unicode_ci	16 KB	-
boletim_clinico	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	80 KB	-
calendario	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	64 KB	-
complemento	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	80 KB	-
consulta	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	80 KB	-
contato	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	latin1_swedish_ci	16 KB	-
dados_clinicos	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	64 KB	-
documentos_pacientes	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
dominio	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	22	InnoDB	latin1_swedish_ci	16 KB	-
empresa	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
especialidade	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	48 KB	-
especialidades	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	5	InnoDB	latin1_swedish_ci	16 KB	-
geografia	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	7	InnoDB	latin1_swedish_ci	112 KB	-
horario_consulta	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2	InnoDB	latin1_swedish_ci	16 KB	-
ilhas	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	16 KB	-

Tabela	Ações	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
login	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	48 KB	-
loginvalidation	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	latin1_swedish_ci	32 KB	-
medicamento	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	48 KB	-
medicamento_receta	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	48 KB	-
medico_especialidade	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	48 KB	-
menu	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	16	InnoDB	latin1_swedish_ci	32 KB	-
menuperfil	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	23	InnoDB	latin1_swedish_ci	48 KB	-
migration	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	1	InnoDB	latin1_swedish_ci	16 KB	-
paciente	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	16 KB	-
pais	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	180	InnoDB	latin1_swedish_ci	16 KB	-
perfil	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	4	InnoDB	latin1_swedish_ci	16 KB	-
receita	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	96 KB	-
resultado_analise	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	latin1_swedish_ci	128 KB	-
tipo_analise	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2	InnoDB	latin1_swedish_ci	48 KB	-
tratamento	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	3	InnoDB	latin1_swedish_ci	80 KB	-
34 tabelas	Soma	534	InnoDB	latin1_swedish_ci	1.6 MB	0 Bytes

Figura 13: Base de Dados do Sistema

CSS - é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML.

```
5
6 .wrap {
7   min-height: 100%;
8   height: auto;
9   margin: 0 auto -60px;
10  padding: 0 0 0px;
11 }
12
13 .wrap > .container {
14   padding: 0px 15px 20px;
15 }
16
17 .footer {
18   background-color: #7f7f7e;
19   border-top: 1px solid #ddd;
20   padding-top: 20px;
21   color: #ffffff;
22   clear: both;
23   margin-top: 100px;
24 }
25
26 .jumbotron {
27   text-align: center;
28   background-color: transparent;
29 }
30
```

Figura 14: Exemplo de um Código CSS

4.2. Descrição das funcionalidades do sistema

O funcionamento do sistema é apresentado com base nos casos de uso descritos, sendo apresentadas telas das aplicações em execução e descrição dos passos para melhor entendimento.

4.2.1. Front-End do sistema

O Front-end é a primeira camada que encontramos quando acedemos a um site, um sistema Web ou uma intranet, por exemplo. Esta camada é responsável pela interface que o utilizador irá encontrar ao navegar pelo sistema, incluindo design da página, a cor, formulários de contato, códigos adicionais, todos os inputs no site, tudo isso é front-end.

A separação entre design, programação e conteúdo permite uma grande flexibilidade na produção de sites com design extremamente personalizados. Além disso, existem milhares de templates prontos que agilizam o processo de criação de sites com rapidez e menor investimento.

O sistema vai estar assento num servidor da empresa, no sistema intranet, será chamado através de um endereço e será aberto a página principal (Home) que tem algumas informações do serviço sem que o paciente esteja cadastrado.

No menu da página principal (Home) encontramos também a página Simarc que tem informações sobre a empresa, logo de seguida encontramos a página Consulta é apresentada nela um calendário onde o paciente faz a sua marcação de consulta, logo que fizer a sua marcação vai direto a página de acesso ao sistema ou a página de login, também temos a página Contato onde encontra um formulario para envios de email.



Figura 15: Tela Página Inicial



Figura 16: Tela Página Simarc

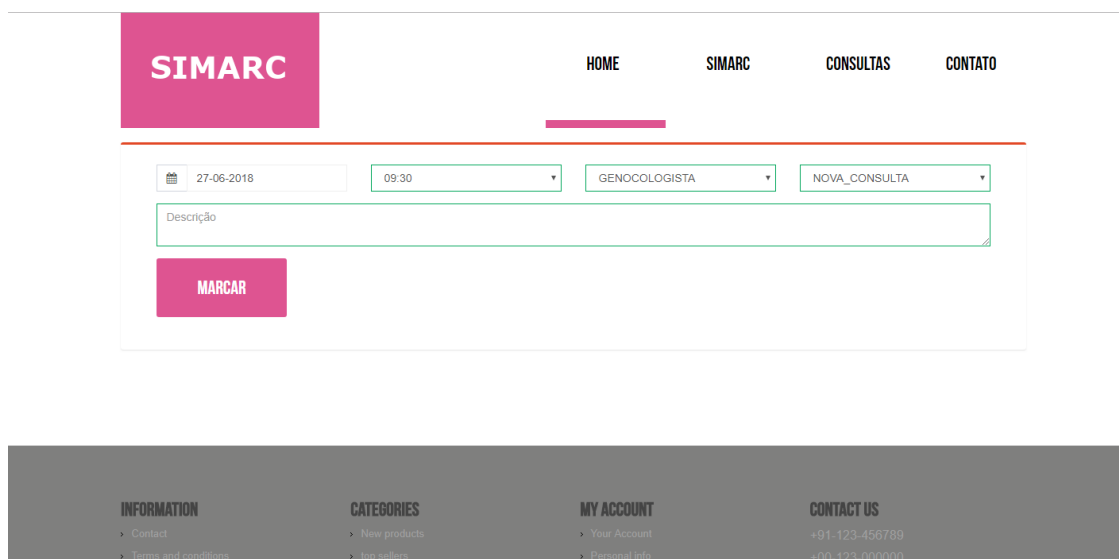


Figura 17: Tela Página Consulta

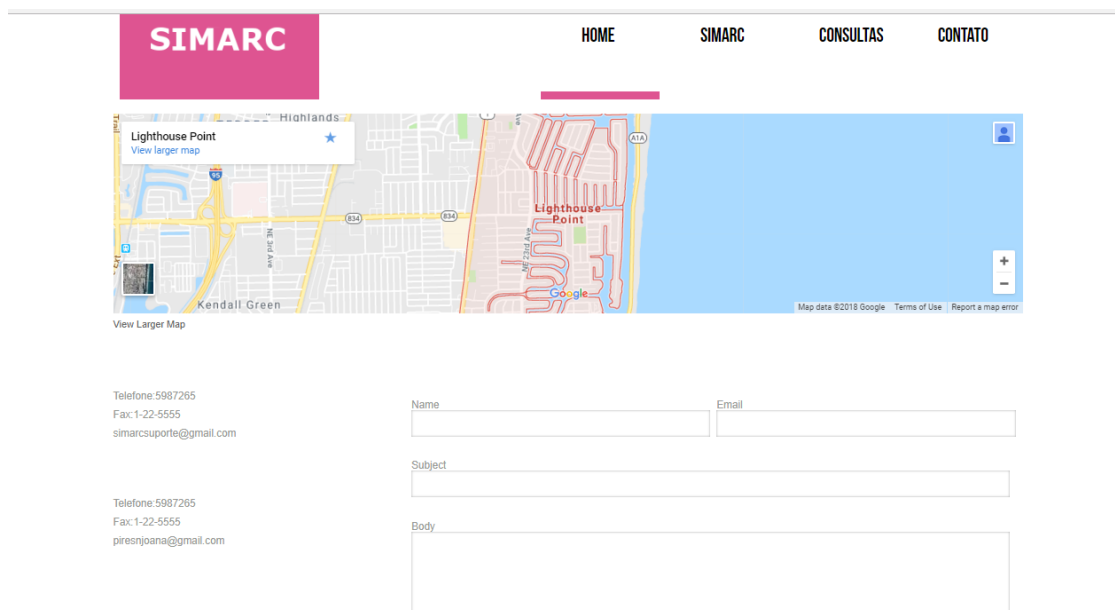


Figura 18: Tela Página Contato

4.2.2. Back-end do sistema

O back-end de um sistema é a parte que o utilizador não consegue ver e nem usar. Mas ela é fundamental para a existência de um site ou sistema. Questões sobre performance e persistência de dados são atividades rotineiras de um programador back-end.

Nesta camada também encontramos alguns códigos *Server-Side*, que funcionam exclusivamente no servidor, onde o utilizador final apenas faz inserção de dados (via front-end), não podendo alterar ou modificar esses códigos.

Interface de acesso ao sistema

Logo que o utilizador acede o sistema, ele encontra a página de login, como pode ser observado na figura. Nesta tela, o utilizador insere seu login e sua senha de acesso para poder aceder ao sistema.

Log In

Email

Palavra-passe

Login

[Recuperar palavra-passe](#)

Figura 19: Tela login do sistema

Caso as credenciais informadas estejam corretas, o utilizador acederá a página contendo um menu personalizado (de acordo com o seu nível de acesso).

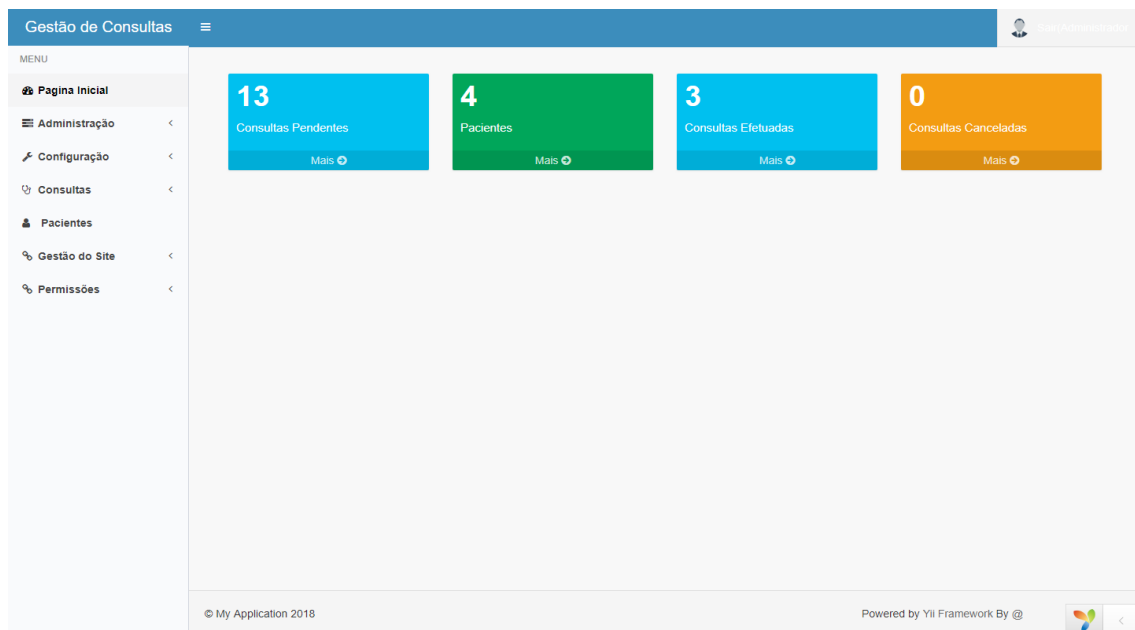


Figura 20: Tela Principal do sistema

Apos fazer login, o sistema verifica se trata de um utilizador normal (funcionário) ou do administrador do sistema, e conforme o caso, mostra o menu específico para o utilizador comum ou para o administrador respetivamente.

Se no caso do utilizador que fizer login for o administrador o sistema apresenta o menu para o administrador.

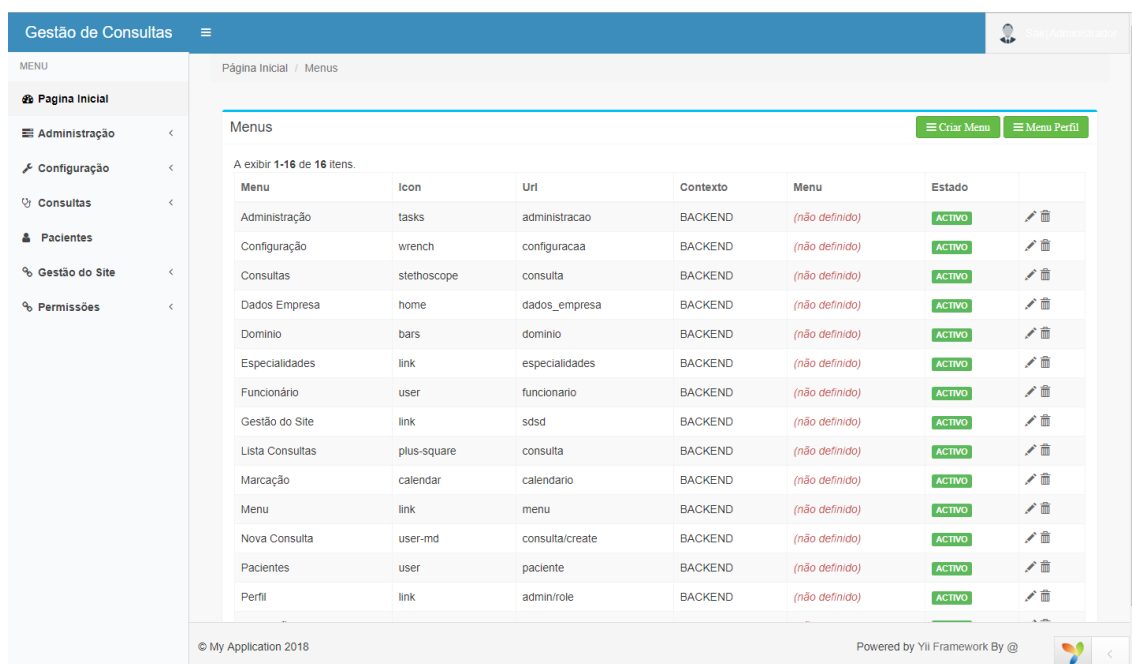


Figura 21: Tela Menus do sistema

Para cadastrar funcionário recebe as diversas informações identificadoras do funcionário (Informações Gerais, Contatos e Endereços, Especialidades).

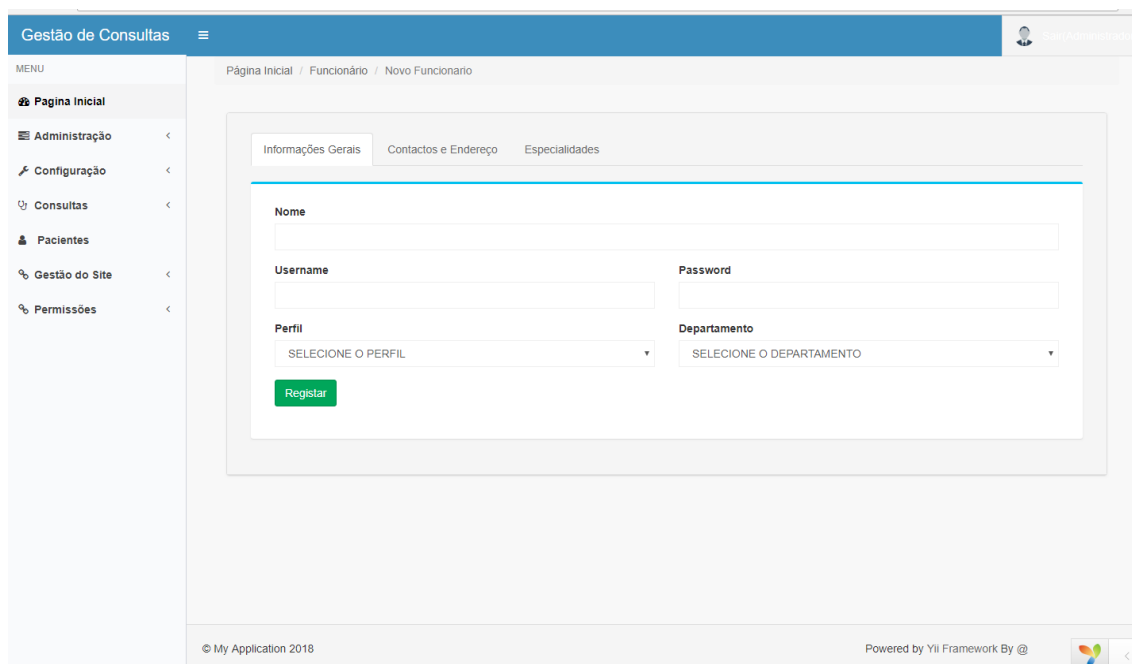


Figura 22: Tela Cadastrar Funcionário do sistema

Para marcar ou agendar uma consulta é disponibilizado um calendário para escolher o dia da consulta.

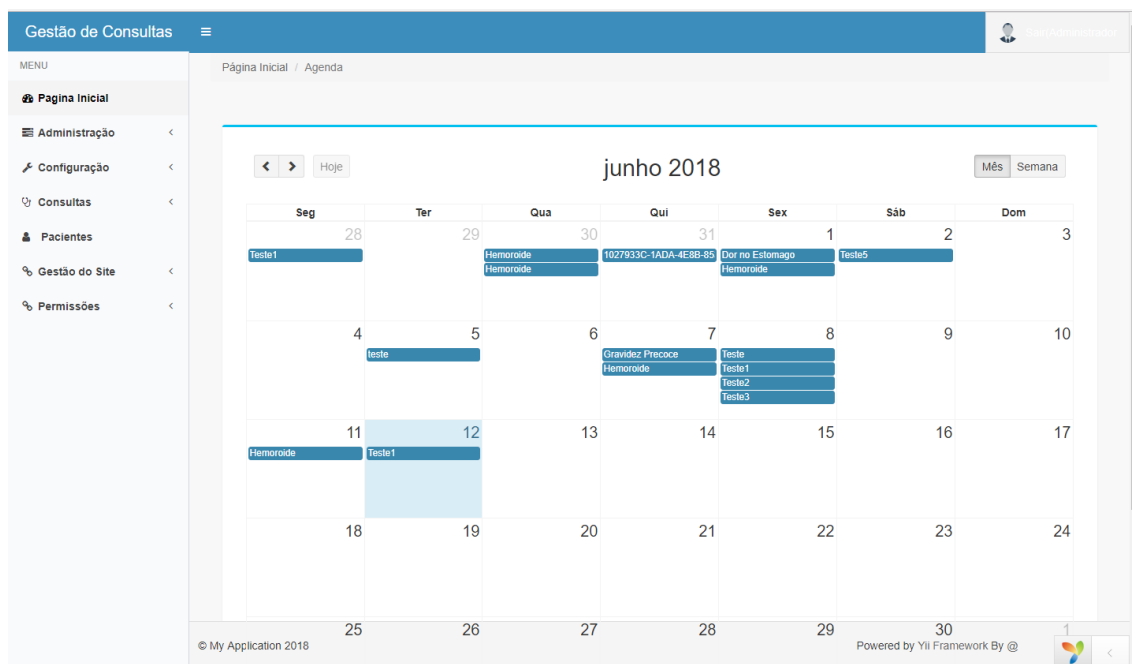


Figura 23: Tela Agendar Consulta no sistema

Apos ter escolhido o dia da consulta será exibido uma caixa onde introduzem os dados para registrar a consulta.

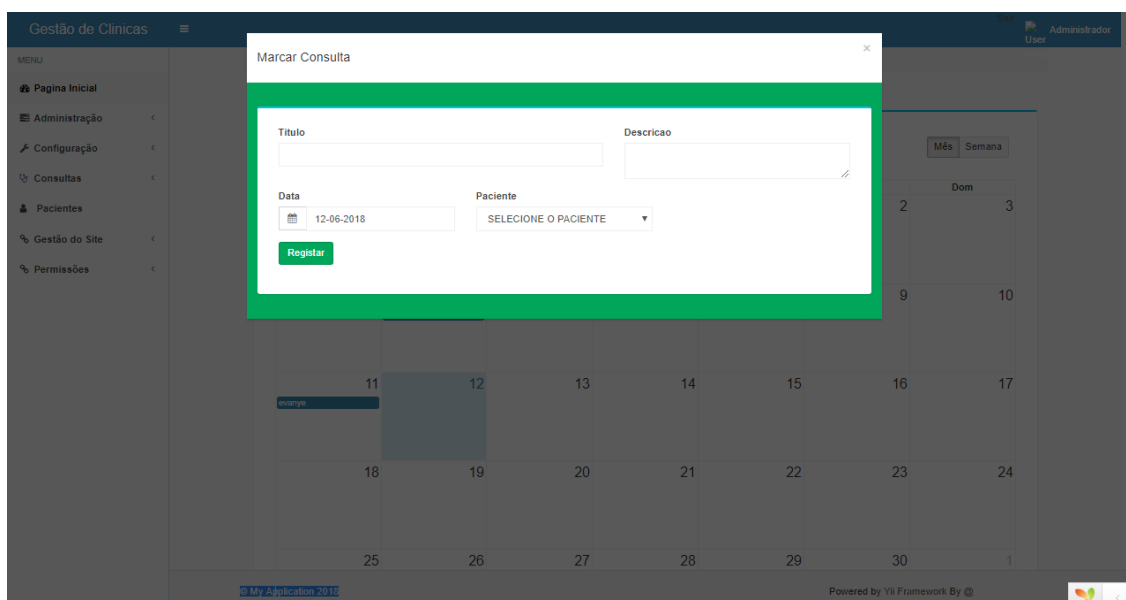


Figura 24: Tela Marcar consulta

Apos ter registrado a consulta o paciente fica em espera.

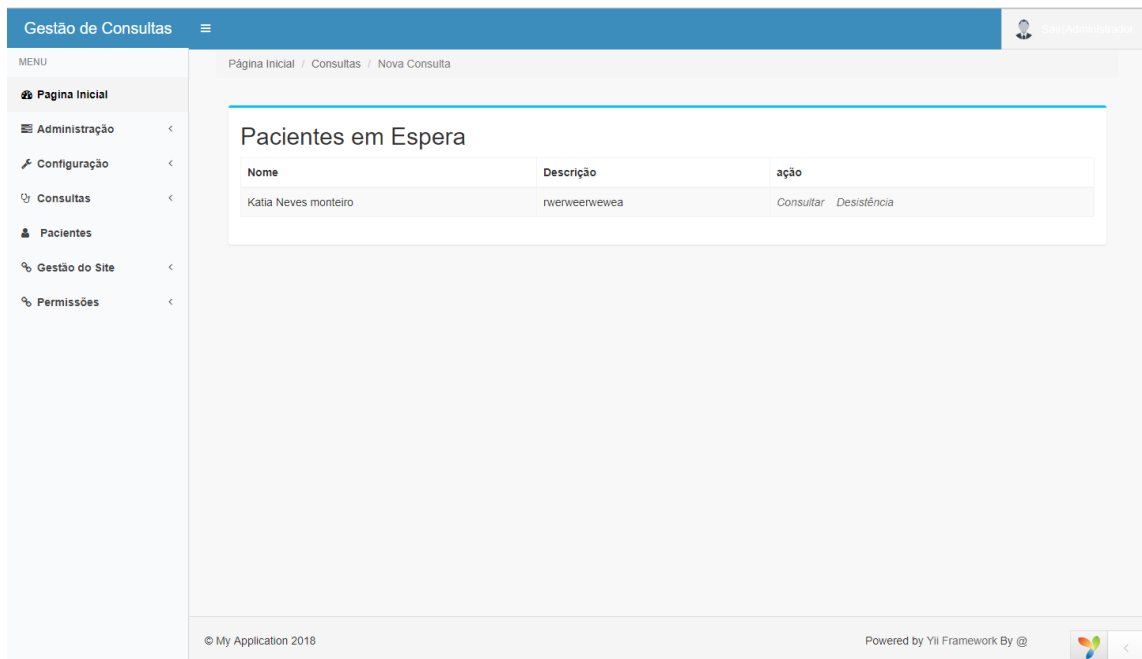


Figura 25: Tela Paciente em espera no sistema

Clica em consultar e abre a tela consulta onde será introduzido os dados (paciente, boletim clínico, tratamento, receita e analise).

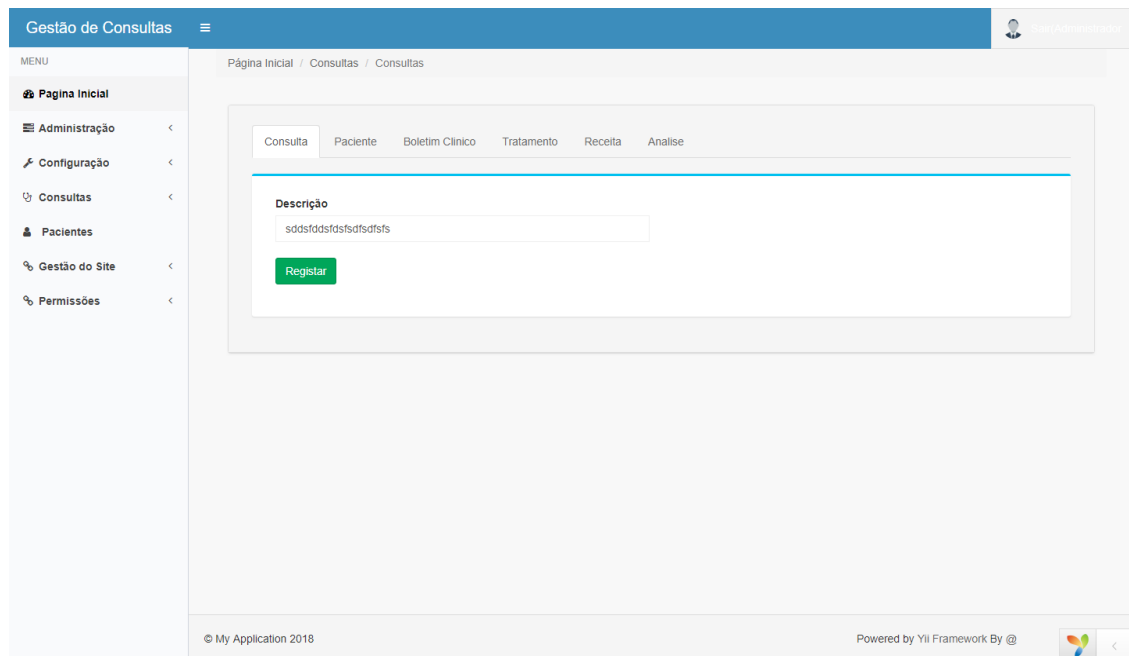


Figura 26: Tela Consulta no sistema

Nesta tela será listada todos os pacientes do sistema.

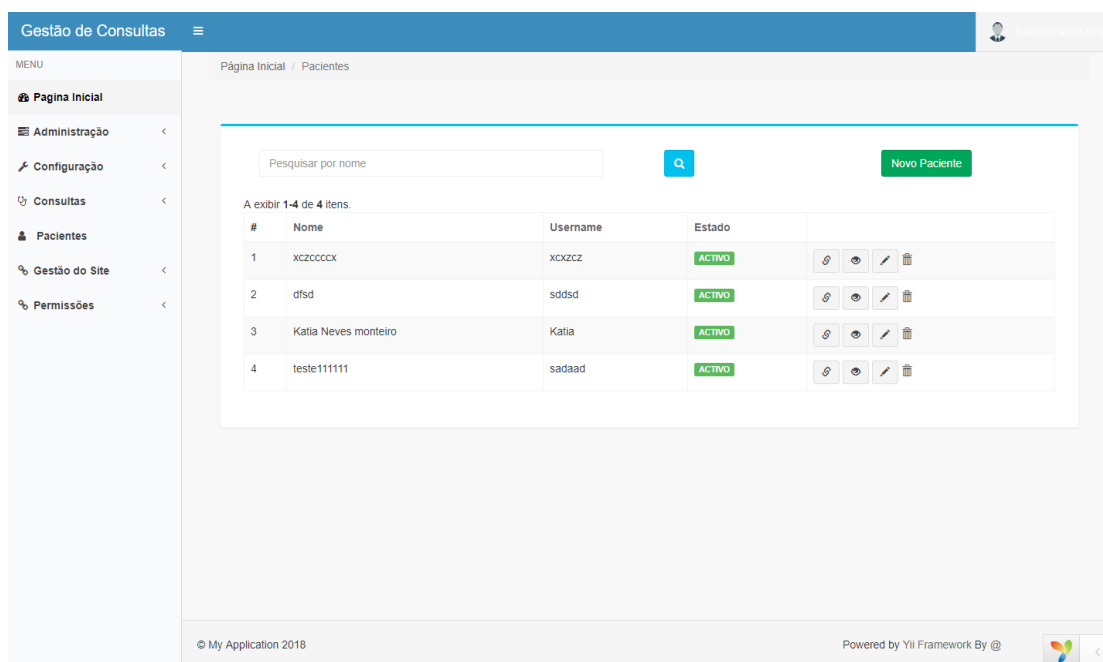


Figura 27: Tela Lista Pacientes do sistema

Se clicar no botão novo paciente aparece a tela cadastrar paciente onde serão introduzidos os dados (Informações Gerais, Dados Pessoais, Endereço e Dados Clínicos).

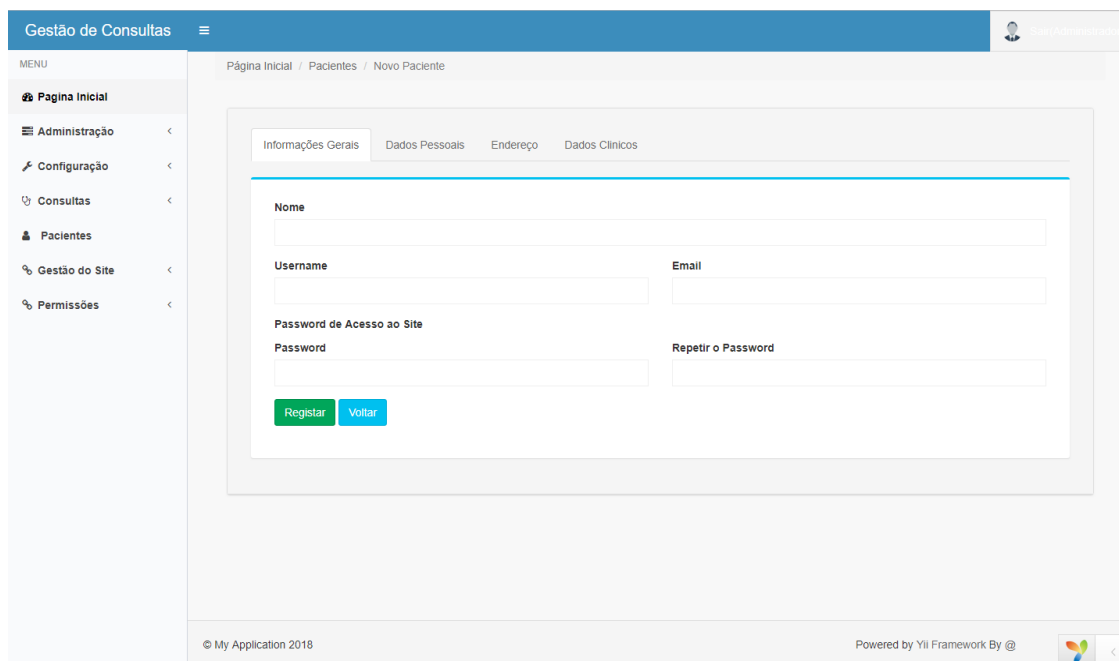


Figura 28: Tela Cadastrar Pacientes no sistema

A tela lista de domínios é onde encontram os dados que são fixos ou valores que possuem determinadas propriedades em comum numa base de dados.

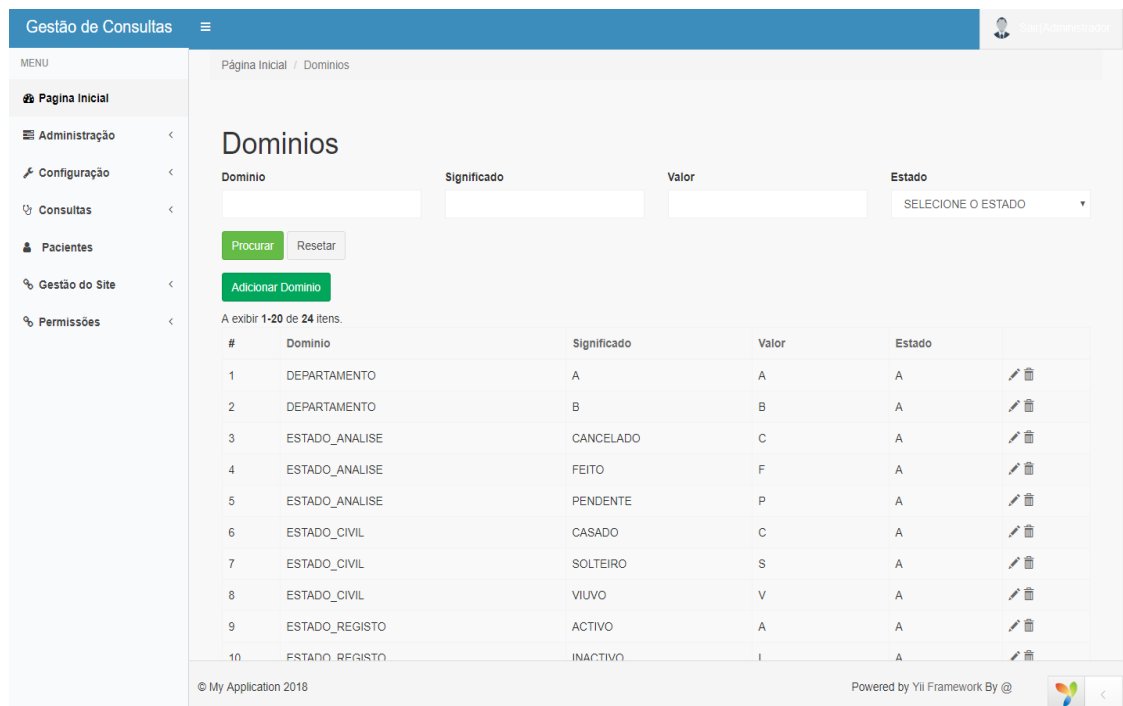


Figura 29: Tela Lista Domínios do sistema

Tela adicionar domínio

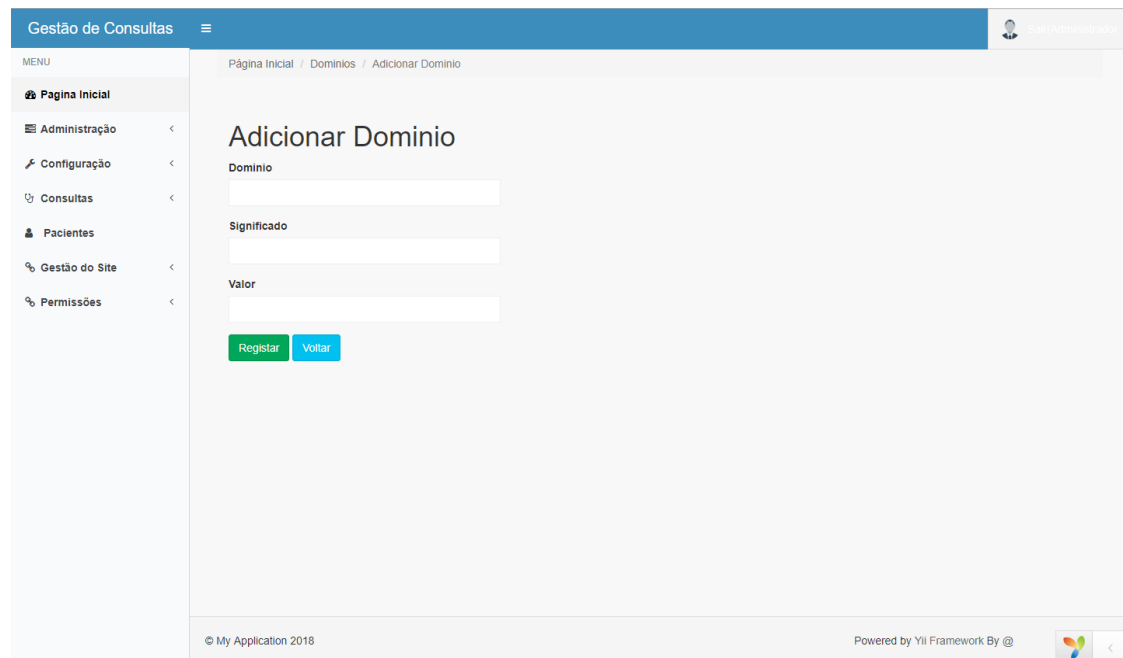


Figura 30: Tela Adicionar Domínios do sistema

Nesta tela encontram as listas de roles ou perfil que fazem parte das permissões numa base de dados.

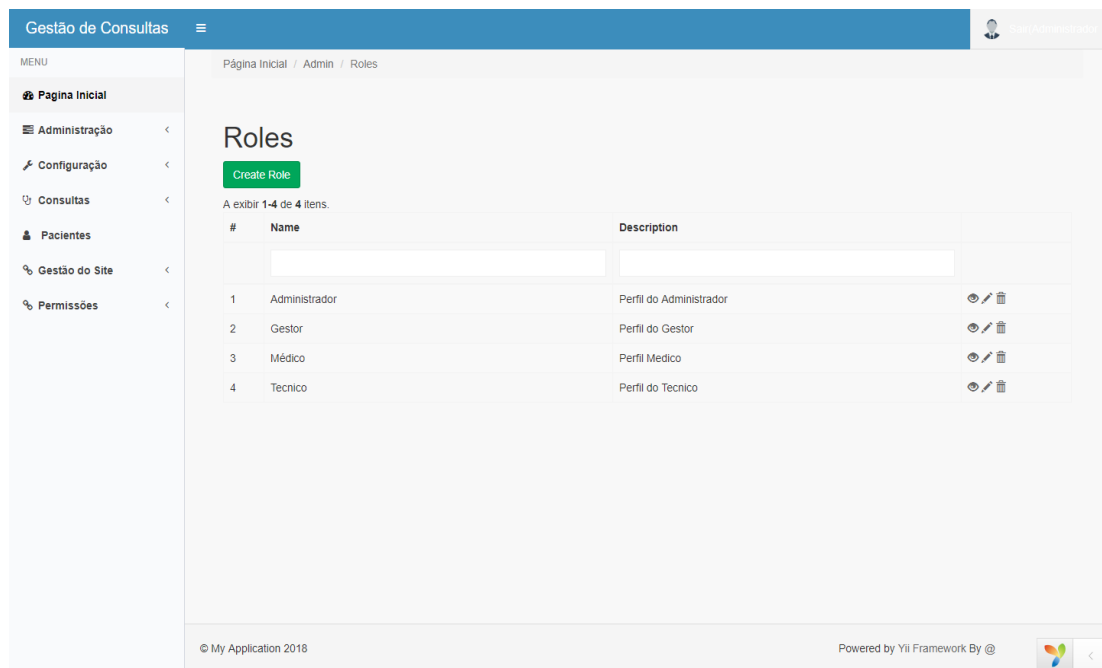


Figura 31: Tela Lista de Role do sistema

Tela para criar role ou perfil.

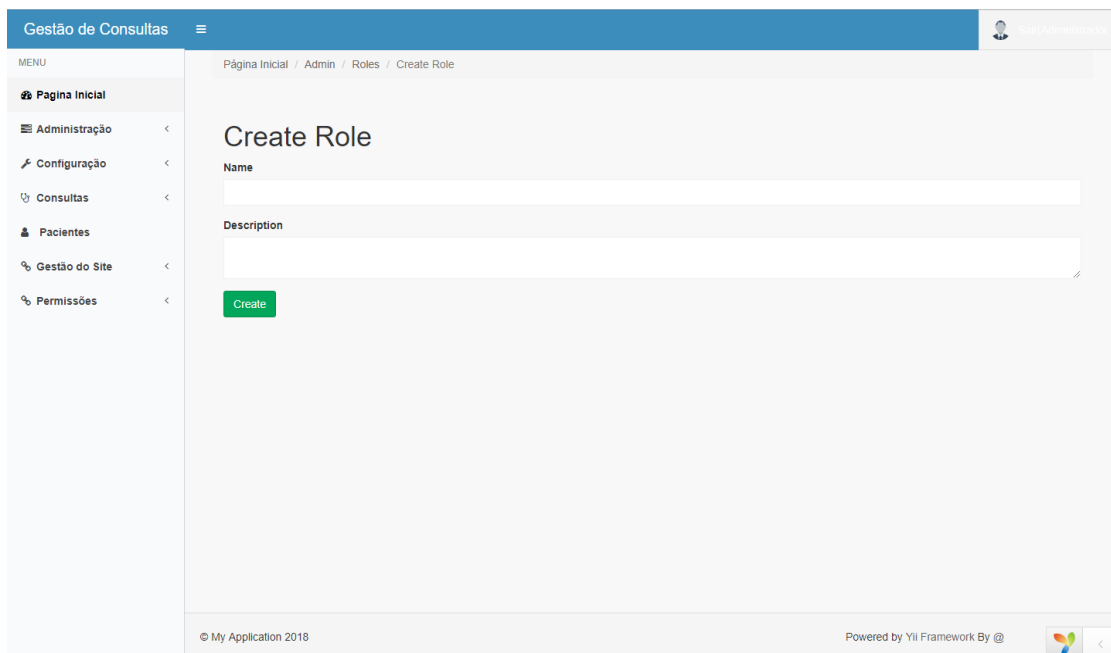


Figura 32: Tela Criar Role do sistema

Tela de routes ou permissões atribuídas a cada perfil.

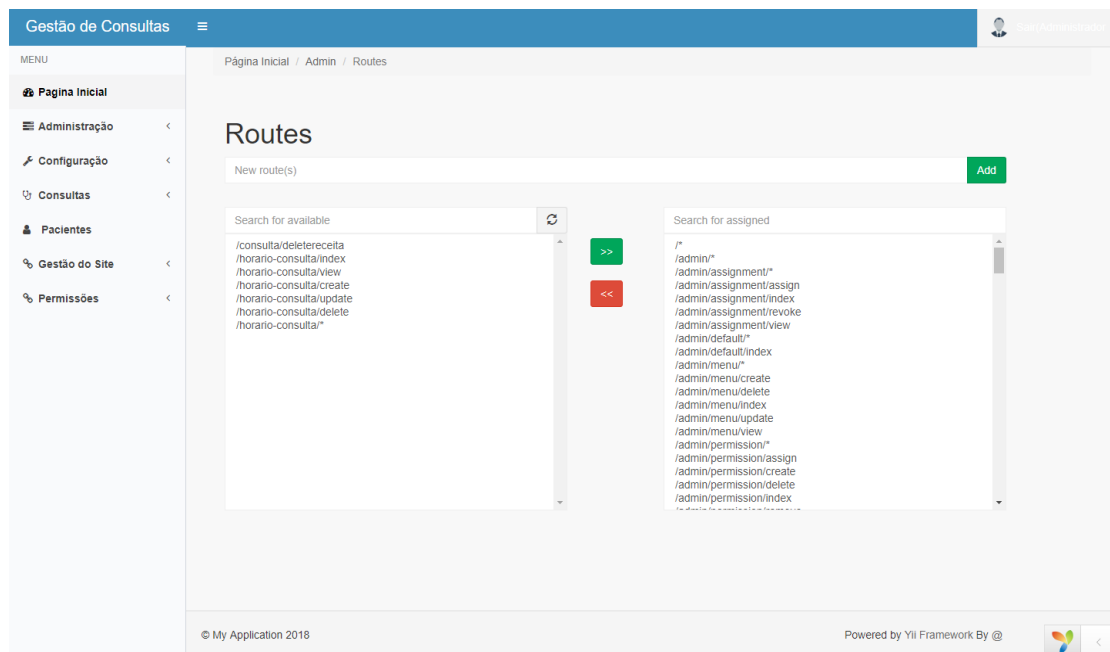


Figura 33: Tela Criar Routes do sistema

CAPITULO V

5. Conclusão

A Medicina moderna, suas organizações e profissionais da saúde convivem diariamente com um grande desafio de gerir grandes quantidades de informações o que se for feito de forma arcaico torna-se impossível corresponder as demandas. O uso das tecnologias nesta área é de extrema importância para que as coisas possam fluir de maneira tranquila, simples e eficaz, dando mais celeridade aos procedimentos realizados diariamente assegurando uma maior recolha e armazenamento de informações, mas também garante uma maior segurança para os dados pessoais dos utentes.

Em observância a tais questões, este trabalho propôs-se o desenvolvimento de um sistema com o objetivo de melhor gerenciar as consultas médicas, de maneira a padronizar e facilitar o controle das atividades do Sistema. Durante o processo de concessão do referido sistema, foram utilizadas várias linguagens de programação, tais como, PHP, HTML, JavaScript e CSS. Procedeu-se de tal modo para extrair das linguagens suas qualidades e, assim, elaborar o sistema de forma satisfatória para atingir o objetivo proposto.

Á priori já sabia de antemão que a elaboração desse projeto não seria fácil devido a complexidade que o mesmo acarretaria, mas, entretanto, nunca desistir, visto que a ambição de concluir o projeto era muito maior do que os obstáculos que seriam transpostos. Depois de ter esse projeto plenamente concluído e pronto para o uso pode-se afirmar que o primeiro objetivo que foi criado que é marcação de consulta e análise online foi alcançado. Porém, para quem irá usufruir do mesmo encontrará uma outra vantagem no que tange a gestão, porque o sistema vai lhes fornecer vários dados numéricas que possibilitará a elaboração de relatórios com vistas ao auxílio para melhor tomadas de decisões futuras para o bem das organizações e estruturas de saúde.

5.1. Dificuldades encontradas

Colocar em prática conhecimentos sobre os sistemas de informação, os aspetos técnicos no que se refere às tecnologias utilizadas, as linguagens de programação para a *Web*, e os

frameworks utilizados, foi uma ótima experiência que supera todas as dificuldades enfrentadas para realização e conclusão deste trabalho.

O aprendizado de um *framework* totalmente desconhecido e o aprofundamento de conhecimentos nas linguagens de programação *Web* proporcionou um esforço maior nas pesquisas na internet e em leituras de livros para a obtenção do conhecimento necessário para o desenvolvimento do sistema.

5.2. Trabalhos Futuros

Como sugestões para trabalhos futuros sugere-se os seguintes:

- Melhorar a interface e criar novos módulos para o Site;
- Disponibilizar um chat, permitindo a conversa instantânea entre pacientes e médicos;
- Melhorar a apresentação e a eficiência do sistema.

Referências Bibliográficas

LOPES, F; M. Morais e A. Carvalho (2005). *Desenvolvimento de Sistema de Informação*. FCA: Editora de Informática [Acesso em: 21/11/2017 às 12:51].

MARQUES, Joaquim e SERRÃO, Carlos (2007). *PHP5*. FCA Editora de Informática [Acesso em: 23/11/2017 às 17:51].

MIGUEL, António (2010). *Gestão de projectos de software*. FCA – Editora de Informática [Acesso em: 22/11/2017 às 16:51].

SILVA, Alberto (2001). *UML Metodologia e Ferramentas*. Editora Centro Atlântico [Acesso em: 21/11/2017 às 16:51].

Acesso a internet

<http://tableless.com.br/entendendo-o-padrao-mvc-na-pratica/>

<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>, [Acesso em: 18/03/18 11:00]

<http://www.devmedia.com.br/php-mvc-aplicando-o-padrao-mvc-no-php-yii-framework/>

https://pt.wikipedia.org/wiki/Diagrama_de_caso_de_uso, [Acesso em: 17/11/17 11:00]

<https://www.webdevbr.com.br/mvc-e-php-entendendo-o-padrao-na-pratica-criando-um-framework-php>

<https://www.youtube.com/>

<http://www.significados.com.br/html/>, [Acesso em: 11/11/17 14:00]

< <http://www.yiiframework.com/> > [Acesso em: 28/11/17 11:00]

<http://www.yiiframework.com/doc-2.1/guide-security-authorization.html#rbac>

<https://adminlte.io/themes/AdminLTE/index2.html>, [Acesso em: 16/11/17 15:30]

<https://adminlte.io/themes/AdminLTE/>, [Acesso em: 16/01/18 14:30]

< <http://www.bsourcecode.com/> > [Acesso em: 15/03/18 11:00]

<https://pt.wikipedia.org/wiki/JQuery>, [Acesso em: 17/03/18 16:00]

<http://www.yiiframework.com/doc/guide/1.1/pt/quickstart.what-is-yii>)

<https://www.yiiframework.com/doc/guide/2.1/pt-br>